# DAQ

# PCI-DIO-96/PXI™-6508/PCI-6503 User Manual

**96-Bit and 24-Bit Parallel Digital I/O Interface for PCI, PXI, and CompactPCI**

NATIONAL INSTRUMENTS™

March 1998 Edition
Part Number 320938C-01

**Internet Support**
E-mail: support@natinst.com
FTP Site: ftp.natinst.com
Web Address: http://www.natinst.com

**Bulletin Board Support**
BBS United States: 512 794 5422
BBS United Kingdom: 01635 551422
BBS France: 01 48 65 15 59

**Fax-on-Demand Support**
512 418 1111

**Telephone Support (USA)**
Tel: 512 795 8248
Fax: 512 794 5678

**International Offices**
Australia 03 9879 5166, Austria 0662 45 79 90 0, Belgium 02 757 00 20, Brazil 011 288 3336,
Canada (Ontario) 905 785 0085, Canada (Québec) 514 694 8521, Denmark 45 76 26 00, Finland 09 725 725 11,
France 01 48 14 24 24, Germany 089 741 31 30, Hong Kong 2645 3186, Israel 03 6120092, Italy 02 413091,
Japan 03 5472 2970, Korea 02 596 7456, Mexico 5 520 2635, Netherlands 0348 433466, Norway 32 84 84 00,
Singapore 2265886, Spain 91 640 0085, Sweden 08 730 49 70, Switzerland 056 200 51 51, Taiwan 02 377 1200,
United Kingdom 01635 523545

**National Instruments Corporate Headquarters**
6504 Bridge Point Parkway  Austin, Texas 78730-5039  USA  Tel: 512 794 0100

© Copyright 1996, 1998 National Instruments Corporation. All rights reserved.

# Important Information

## Warranty

The PCI-DIO-96, PXI-6508, and PCI-6503 are warranted against defects in materials and workmanship for a period of one year from the date of shipment, as evidenced by receipts or other documentation. National Instruments will, at its option, repair or replace equipment that proves to be defective during the warranty period. This warranty includes parts and labor.

The media on which you receive National Instruments software are warranted not to fail to execute programming instructions, due to defects in materials and workmanship, for a period of 90 days from date of shipment, as evidenced by receipts or other documentation. National Instruments will, at its option, repair or replace software media that do not execute programming instructions if National Instruments receives notice of such defects during the warranty period. National Instruments does not warrant that the operation of the software shall be uninterrupted or error free.

A Return Material Authorization (RMA) number must be obtained from the factory and clearly marked on the outside of the package before any equipment will be accepted for warranty work. National Instruments will pay the shipping costs of returning to the owner parts which are covered by warranty.

National Instruments believes that the information in this manual is accurate. The document has been carefully reviewed for technical accuracy. In the event that technical or typographical errors exist, National Instruments reserves the right to make changes to subsequent editions of this document without prior notice to holders of this edition. The reader should consult National Instruments if errors are suspected. In no event shall National Instruments be liable for any damages arising out of or related to this document or the information contained in it.

EXCEPT AS SPECIFIED HEREIN, NATIONAL INSTRUMENTS MAKES NO WARRANTIES, EXPRESS OR IMPLIED, AND SPECIFICALLY DISCLAIMS ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. CUSTOMER'S RIGHT TO RECOVER DAMAGES CAUSED BY FAULT OR NEGLIGENCE ON THE PART OF NATIONAL INSTRUMENTS SHALL BE LIMITED TO THE AMOUNT THERETOFORE PAID BY THE CUSTOMER. NATIONAL INSTRUMENTS WILL NOT BE LIABLE FOR DAMAGES RESULTING FROM LOSS OF DATA, PROFITS, USE OF PRODUCTS, OR INCIDENTAL OR CONSEQUENTIAL DAMAGES, EVEN IF ADVISED OF THE POSSIBILITY THEREOF. This limitation of the liability of National Instruments will apply regardless of the form of action, whether in contract or tort, including negligence. Any action against National Instruments must be brought within one year after the cause of action accrues. National Instruments shall not be liable for any delay in performance due to causes beyond its reasonable control. The warranty provided herein does not cover damages, defects, malfunctions, or service failures caused by owner's failure to follow the National Instruments installation, operation, or maintenance instructions; owner's modification of the product; owner's abuse, misuse, or negligent acts; and power failure or surges, fire, flood, accident, actions of third parties, or other events outside reasonable control.

## Copyright

Under the copyright laws, this publication may not be reproduced or transmitted in any form, electronic or mechanical, including photocopying, recording, storing in an information retrieval system, or translating, in whole or in part, without the prior written consent of National Instruments Corporation.

## Trademarks

ComponentWorks™, CVI™, LabVIEW™, Measure™, MITE™, NI-DAQ™, PXI™, SCXI™, and VirtualBench™ are trademarks of National Instruments Corporation.

Product and company names listed are trademarks or trade names of their respective companies.

## WARNING REGARDING MEDICAL AND CLINICAL USE OF NATIONAL INSTRUMENTS PRODUCTS

National Instruments products are not designed with components and testing intended to ensure a level of reliability suitable for use in treatment and diagnosis of humans. Applications of National Instruments products involving medical or clinical treatment can create a potential for accidental injury caused by product failure, or by errors on the part of the user or application designer. Any use or application of National Instruments products for or involving medical or clinical treatment must be performed by properly trained and qualified medical personnel, and all traditional medical safeguards, equipment, and procedures that are appropriate in the particular situation to prevent serious injury or death should always continue to be used when National Instruments products are being used. National Instruments products are NOT intended to be a substitute for any form of established process, procedure, or equipment used to monitor or safeguard human health and safety in medical or clinical treatment.

# Contents

## About This Manual


Organization of This Manual ..... ix
Conventions Used in This Manual ..... x
National Instruments Documentation ..... xi
Related Documentation ..... xii
Customer Communication ..... xii


## Chapter 1
## Introduction


About Your Board ..... 1-1
Using PXI with CompactPCI ..... 1-2
What You Need to Get Started ..... 1-2
Software Programming Choices ..... 1-3
    National Instruments Application Software ..... 1-3
    NI-DAQ Driver Software ..... 1-3
    Register-Level Programming ..... 1-5
Optional Equipment ..... 1-5
Custom Cabling ..... 1-6
Unpacking ..... 1-6


## Chapter 2
## Installation and Configuration


Software Installation ..... 2-1
Hardware Installation ..... 2-1
    Installing the PCI-DIO-96 or PCI-6503 ..... 2-1
    Installing the PXI-6508 ..... 2-2
Board Configuration ..... 2-2


## Chapter 3
## Signal Connections


I/O Connector (PCI-DIO-96, PXI-6508) ..... 3-1
    I/O Connector Pin Assignments ..... 3-1
    Cable Assembly Connectors ..... 3-3
    I/O Connector Signal Descriptions ..... 3-5
I/O Connector (PCI-6503) ..... 3-7
    PCI-6503 I/O Connector Pin Descriptions ..... 3-7
Port C Pin Assignments ..... 3-9

Digital I/O Signal Connections ..... 3-10
Power Connections ..... 3-12
Digital I/O Power-up State Selection ..... 3-12
High DIO Power-up State ..... 3-12
Low DIO Power-up State (PXI-6508, PCI-6503 Only) ..... 3-14
Timing Specifications ..... 3-15
Mode 1 Input Timing ..... 3-17
Mode 1 Output Timing ..... 3-18
Mode 2 Bidirectional Timing ..... 3-19

## Chapter 4
## Theory of Operation

Functional Overview ..... 4-1
PCI Interface Circuitry ..... 4-2
82C55A Programmable Peripheral Interface ..... 4-3
82C53 Programmable Interval Timer (PCI-DIO-96, PXI-6508 Only) ..... 4-3
Interrupt Control Circuitry ..... 4-3

## Appendix A
## Specifications

## Appendix B
## Register-Level Programming

Register Map and Description ..... B-1
Introduction ..... B-1
Register Map ..... B-2
Register Descriptions ..... B-3
Register Description Format ..... B-3
Register Description for the 82C55A ..... B-3
Register Description for the 82C53 (PCI-DIO-96, PXI-6508 Only) ..... B-5
Register Description for the Interrupt Control Registers ..... B-6
Interrupt Control Register 1 ..... B-7
Interrupt Control Register 2 ..... B-9
Interrupt Clear Register (PCI-DIO-96, PXI-6508 Only) ..... B-10
Programming ..... B-10
PCl Local Bus ..... B-10
Programming Examples ..... B-11
PCI Initialization for the PC ..... B-11
PCI Initialization for the Macintosh (PCI-DIO-96, PCI-6503 Only) ..... B-13

Programming Considerations for the 82C55A ..... B-15
Modes of Operation ..... B-15
Mode 0–Basic I/O ..... B-16
Mode 1–Strobed Input ..... B-18
Port C Status-Word Bit Definitions for Input (Ports A and B) ..... B-20
Mode 1–Strobed Output ..... B-22
Port C Status-Word Bit Definitions for Output (Ports A and B) ..... B-24
Mode 2–Bidirectional Bus ..... B-26
Port C Status-Word Bit Definitions for Bidirectional Data Path (Port A Only) ..... B-28
Interrupt Handling ..... B-31
Interrupt Programming Examples for the 82C55A ..... B-31
Programming Considerations for the 82C53 ..... B-32
General Information ..... B-33
Interrupt Programming Example ..... B-33

## Appendix C
## MSM82C55A Data Sheet

## Appendix D
## MSM82C53 Data Sheet

## Appendix E
## Customer Communication

## Glossary

## Index

## Figures

Figure 1-1. The Relationship between the Programming Environment, NI-DAQ, and Your Hardware ..... 1-4

Figure 3-1. PCI-DIO-96 and PXI-6508 Connector Pin Assignments ..... 3-2
Figure 3-2. Cable-Assembly Connector Pinout for Pins 1 through 50 with the R1005050 Ribbon Cable (PCI-DIO-96 and PXI-6508) ..... 3-3
Figure 3-3. Cable-Assembly Connector Pinout for Pins 51 through 100 with the R1005050 Ribbon Cable (PCI-DIO-96 and PXI-6508) ..... 3-4
Figure 3-4. PCI-6503 I/O Connector Pin Assignments ..... 3-7

Figure 3-5. Digital I/O Connections Block Diagram ..... 3-11
Figure 3-6. DIO Channel Configured for High DIO Power-up State with External Load ..... 3-13
Figure 3-7. DIO Channel Configured for Low DIO Power-up State with External Load ..... 3-14
Figure 3-8. Timing Specifications for Mode 1 Input Transfer ..... 3-17
Figure 3-9. Timing Specifications for Mode 1 Output Transfer ..... 3-18
Figure 3-10. Timing Specifications for Mode 2 Bidirectional Transfer ..... 3-19

Figure 4-1. Block Diagram ..... 4-2
Figure 4-2. Interrupt Control Circuitry Block Diagram ..... 4-4

Figure B-1. Control Word Formats for the 82C55A ..... B-4
Figure B-2. Control Word Format for the 82C53 ..... B-5
Figure B-3. Control Word to Configure Port A for Mode 1 Input ..... B-18
Figure B-4. Control Word to Configure Port B for Mode 1 Input ..... B-19
Figure B-5. Port C Pin Assignments on I/O Connector when Port C Configured for Mode 1 Input ..... B-22
Figure B-6. Control Word to Configure Port A for Mode 1 Output ..... B-23
Figure B-7. Control Word to Configure Port B for Mode 1 Output ..... B-23
Figure B-8. Port C Pin Assignments on I/O Connector when Port C Configured for Mode 1 Output ..... B-26
Figure B-9. Control Word to Configure Port A as Mode 2 Bidirectional Data Bus ..... B-27
Figure B-10. Port C Pin Assignments on I/O Connector when Port C is Configured for Mode 2 ..... B-30

## Tables

Table 3-1. Signal Descriptions for PCI-DIO-96 and PXI-6508 I/O Connectors ..... 3-5
Table 3-2. PCI-6503 Signal Descriptions ..... 3-8
Table 3-3. Port C Signal Assignments ..... 3-9
Table 3-4. Signal Names Used in Timing Diagrams ..... 3-16

Table B-1. Register Address Map ..... B-2
Table B-2. Port C Set/Reset Control Words ..... B-5
Table B-3. Mode 0 I/O Configurations ..... B-17

# About This Manual

This manual describes the electrical and mechanical aspects of the PCI-DIO-96, PXI-6508, and PCI-6503 and contains information concerning their installation, operation, and programming. The PCI-DIO-96 and PCI-6503 are members of the National Instruments PCI Series of expansion boards for PCI bus computers. The PXI-6508 is a member of the National Instruments PXI family of expansion boards for PXI and CompactPCI chassis. These boards are designed for high-performance data acquisition and control for applications in laboratory testing, production testing, and industrial process monitoring and control.

## Organization of This Manual

The *PCI-DIO-96/PXI-6508/PCI-6503 User Manual* is organized as follows:

- Chapter 1, *Introduction*, describes the PCI-DIO-96, PXI-6508, and PCI-6503; lists what you need to get started, software programming choices, and optional equipment; describes custom cabling options; and explains how to unpack your board.
- Chapter 2, *Installation and Configuration*, describes how to install and configure your PCI-DIO-96, PXI-6508, or PCI-6503 board.
- Chapter 3, *Signal Connections*, describes how to make input and output signal connections to your PCI-DIO-96, PXI-6508, and PCI-6503 via the board I/O connector.
- Chapter 4, *Theory of Operation*, contains a functional overview of the PCI-DIO-96, PXI-6508, and PCI-6503 and explains the operation of each functional unit.
- Appendix A, *Specifications*, lists the specifications for the PCI-DIO-96, PXI-6508, and PCI-6503.
- Appendix B, *Register-Level Programming*, describes in detail the address and function of each PCI-DIO-96, PXI-6508, and PCI-6503 register, contains instructions on how to operate DIO board circuitry, and provides examples of the programming steps necessary to execute an operation.

- Appendix C, *MSM82C55A Data Sheet*, contains a manufacturer data sheet for the MSM82C55A CMOS programmable peripheral interface from OKI Semiconductor. This interface is used on the PCI-DIO-96, PXI-6508, and PCI-6503.
- Appendix D, *MSM82C53 Data Sheet*, contains a manufacturer data sheet for the MSM82C53 CMOS programmable interval timer from OKI Semiconductor. This timer is used on the PCI-DIO-96 and PXI-6508.
- Appendix E, *Customer Communication*, contains forms you can use to request help from National Instruments or to comment on our products.
- The *Glossary* contains an alphabetical list and description of terms used in this manual, including abbreviations, acronyms, metric prefixes, mnemonics, symbols, and terms.
- The *Index* contains an alphabetical list of key terms and topics used in this manual, including the page where each one can be found.

# Conventions Used in This Manual

The following conventions are used in this manual:

| | |
|---|---|
| < > | Angle brackets containing numbers separated by an ellipses represent a range of values associated with a bit, signal, or port (for example, ACH<0..7> stands for ACH0 through ACH7). |
| ☞ | This icon to the left of bold italicized text denotes a note, which alerts you to important information. |
| ⚠ | This icon to the left of bold italicized text denotes a caution, which advises you of precautions to take to avoid injury, data loss, or a system crash. |
| **bold** | Bold text denotes menu items, function panel items, and dialog box buttons or options. |
| ***bold italic*** | Bold italic text denotes a note or caution. |
| *italic* | Italic text denotes emphasis, a cross reference, or an introduction to a key concept. This font also denotes text from which you supply the appropriate word or value, as in Windows 3.*x*. |
| Macintosh | Macintosh refers to all Macintosh computers with PCI bus, unless otherwise noted. |

| | |
|---|---|
| `monospace` | Text in this font denotes text or characters that are to be literally input from the keyboard, sections of code, programming examples, and syntax examples. This font is also used for the proper names of disk drives, paths, directories, programs, subprograms, subroutines, device names, functions, variables, parameters, file names, and extensions, and for statements and comments taken from program code. |
| NI-DAQ | NI-DAQ is used in this manual to refer to NI-DAQ software for PC or Macintosh computers unless otherwise noted. |
| PC | PC refers to all IBM PC compatible computers with PCI bus, unless otherwise noted. |
| PPI *x* | PPI *x*, where the *x* is replaced by A, B, C, or D, refers to one of the four programmable peripheral interface (PPI) chips on the PCI-DIO-96 or PXI-6508. The PCI-6503 contains only one PPI, PPIA. |
| SCXI | SCXI stands for Signal Conditioning eXtensions for Instrumentation and is a National Instruments product line designed to perform front-end signal conditioning for National Instruments plug-in DAQ boards. |
| Your DIO board | Your DIO board refers to either the PCI-DIO-96, PXI-6508, or PCI-6503 board. |

# National Instruments Documentation

The *PCI-DIO-96/PXI-6508/PCI-6503 User Manual* is one piece of the documentation set for your data acquisition system. You could have any of several types of manuals, depending on the hardware and software in your system. Use the manuals you have as follows:

- *Getting Started with SCXI*—If you are using SCXI, this is the first manual you should read. It gives an overview of the SCXI system and contains the most commonly needed information for the modules, chassis, and software.
- Your SCXI hardware user manuals—If you are using SCXI, read these manuals next for detailed information about signal connections and module configuration. They also explain in greater detail how the module works and contain application hints.
- Your DAQ hardware user manuals—These manuals have detailed information about the DAQ hardware that plugs into or is connected to your computer. Use these manuals for hardware installation and configuration instructions, specification information about your DAQ hardware, and application hints.

- Software documentation—Examples of software documentation you may have are the LabVIEW, LabWindows/CVI, or ComponentWorks documentation sets and the NI-DAQ documentation. After you set up your hardware system, use either the application software or the NI-DAQ documentation to help you write your application. If you have a large and complicated system, it is worthwhile to look through the software documentation before you configure your hardware.
- Accessory installation guides or manuals—If you are using accessory products, read the terminal block and cable assembly installation guides or accessory board user manuals. They explain how to physically connect the relevant pieces of the system. Consult these guides when you are making your connections.
- *SCXI Chassis User Manual*—If you are using SCXI, read these manuals for maintenance information on the chassis and installation instructions.

# Related Documentation

The following National Instruments document contains information that you may find helpful as you read this manual:

- Application Note 025, *Field Wiring and Noise Considerations for Analog Signals*

The following documents also contain information that you may find helpful as you read this manual:

- Your computer's technical reference manual
- *PCI Local Bus Specification, Revision 2.1*
- National Instruments *PXI Specification, Revision 1.0*
- *PICMG 2.0 R2.1 CompactPCI*

# Customer Communication

National Instruments wants to receive your comments on our products and manuals. We are interested in the applications you develop with our products, and we want to help if you have problems with them. To make it easy for you to contact us, this manual contains comment and configuration forms for you to complete. These forms are in Appendix E, *Customer Communication*, at the end of this manual.

# 1 Introduction

This chapter describes the PCI-DIO-96, PXI-6508, and PCI-6503; lists what you need to get started, software programming choices, and optional equipment; describes custom cabling options; and explains how to unpack your board.

## About Your Board

Thank you for purchasing a National Instruments PCI-DIO-96, PXI-6508, or PCI-6503 board. The PCI-DIO-96 is a 96-bit, parallel, digital I/O interface for PCI bus computers. The PXI-6508 is a 96-bit, parallel, digital I/O interface for PXI and CompactPCI chassis. The PCI-6503 is a 24-bit, parallel, digital I/O interface for PCI bus computers.

Four 82C55A programmable peripheral interface (PPI) chips control the 96 bits of TTL-compatible digital I/O on the PCI-DIO-96 or PXI-6508. On the PCI-6503, one 82C55A PPI controls the 24 bits of TTL-compatible digital I/O. The OKI Semiconductor 82C55A PPI chips can operate in unidirectional mode, bidirectional mode, or handshaking mode and can generate interrupt requests to your computer. The digital I/O lines are all accessible through a 100-pin female connector on the PCI-DIO-96 or PXI-6508 and a 50-pin male connector on the PCI-6503.

Your DIO board is a completely switchless and jumperless DAQ board. All resource allocation is completed automatically at startup, so you will not need to set interrupt levels or base addresses.

With your DIO board, you can use your computer as a digital I/O system controller for laboratory testing, production testing, and industrial process monitoring and control.

Detailed PCI-DIO-96, PXI-6508, and PCI-6503 specifications are in Appendix A, *Specifications*.

# Using PXI with CompactPCI

Using PXI-compatible products with standard CompactPCI products is an important feature provided by the *PXI Specification*, Revision 1.0. If you use a PXI-compatible plug-in device in a standard CompactPCI chassis, you will be unable to use PXI-specific functions, but you can still use the basic plug-in device functions.

The CompactPCI specification permits vendors to develop sub-buses that coexist with the basic PCI interface on the CompactPCI bus. Compatible operation is not guaranteed between CompactPCI devices with different sub-buses nor between CompactPCI devices with sub-buses and PXI. The standard implementation for CompactPCI does not include these sub-buses. Your PXI-6508 device will work in any standard CompactPCI chassis adhering to the *PICMG 2.0 R2.1 CompactPCI* core specification.

# What You Need to Get Started

To set up and use your PCI-DIO-96/PXI-6508 or PCI-6503 board, you will need the following:

- ❑ PCI-DIO-96, PXI-6508, or PCI-6503 board
- ❑ *PCI-DIO-96/PXI-6508/PCI-6503 User Manual*
- ❑ One of the following software packages and documentation:
  - ComponentWorks
  - LabVIEW for Macintosh
  - LabVIEW for Windows
  - LabWindows/CVI for Windows
  - NI-DAQ for Macintosh
  - NI-DAQ for PC Compatibles
  - Measure
  - VirtualBench
- ❑ Your computer, or PXI or CompactPCI chassis and controller

# Software Programming Choices

There are several options to choose from when programming your National Instruments DAQ hardware. You can use LabVIEW, LabWindows/CVI, ComponentWorks, or other application development environments with the NI-DAQ instrument driver, or you can register-level program.

## National Instruments Application Software

ComponentWorks contains tools for data acquisition and instrument control built on NI-DAQ driver software. ComponentWorks provides a higher-level programming interface for building virtual instruments through standard OLE controls and DLLs. With ComponentWorks, you can use all of the configuration tools, resource management utilities, and interactive control utilities included with NI-DAQ.

LabVIEW features interactive graphics, a state-of-the-art user interface, and a powerful graphical programming language. The LabVIEW Data Acquisition VI Library, a series of VIs for using LabVIEW with National Instruments DAQ hardware, is included with LabVIEW. The LabVIEW Data Acquisition VI Library is functionally equivalent to the NI-DAQ software.

LabWindows/CVI features interactive graphics, a state-of-the-art user interface, and uses the ANSI standard C programming language. The LabWindows/CVI Data Acquisition Library, a series of functions for using LabWindows/CVI with National Instruments DAQ hardware, is included with the NI-DAQ software kit. The LabWindows/CVI Data Acquisition Library is functionally equivalent to the NI-DAQ software.

Using ComponentWorks, LabVIEW, or LabWindows/CVI software will greatly reduce the development time for your data acquisition and control application.

## NI-DAQ Driver Software

The NI-DAQ driver software is included at no charge with all National Instruments DAQ hardware. NI-DAQ is not included with SCXI or accessory products, except the SCXI-1200. NI-DAQ has an extensive library of functions that you can call from your application programming environment. These functions include routines for analog input (A/D conversion), buffered data acquisition (high-speed A/D conversion), analog output (D/A conversion), waveform generation (timed D/A

conversion), digital I/O, counter/timer operations, SCXI, RTSI, self-calibration, messaging, and acquiring data to memory.

NI-DAQ has both high-level DAQ I/O functions for maximum ease of use and low-level DAQ I/O functions for maximum flexibility and performance. Examples of high-level functions are streaming data to disk or acquiring a certain number of data points. An example of a low-level function is writing directly to registers on the DAQ device. NI-DAQ does not sacrifice the performance of National Instruments DAQ devices because it lets multiple devices operate at their peak performance.

NI-DAQ also internally addresses many of the complex issues between the computer and the DAQ hardware such as programming interrupts and DMA controllers. NI-DAQ maintains a consistent software interface among its different versions so that you can change platforms with minimal modifications to your code. Whether you are using conventional programming languages, ComponentWorks, LabVIEW, or LabWindows/CVI, your application uses the NI-DAQ driver software, as illustrated in Figure 1-1.

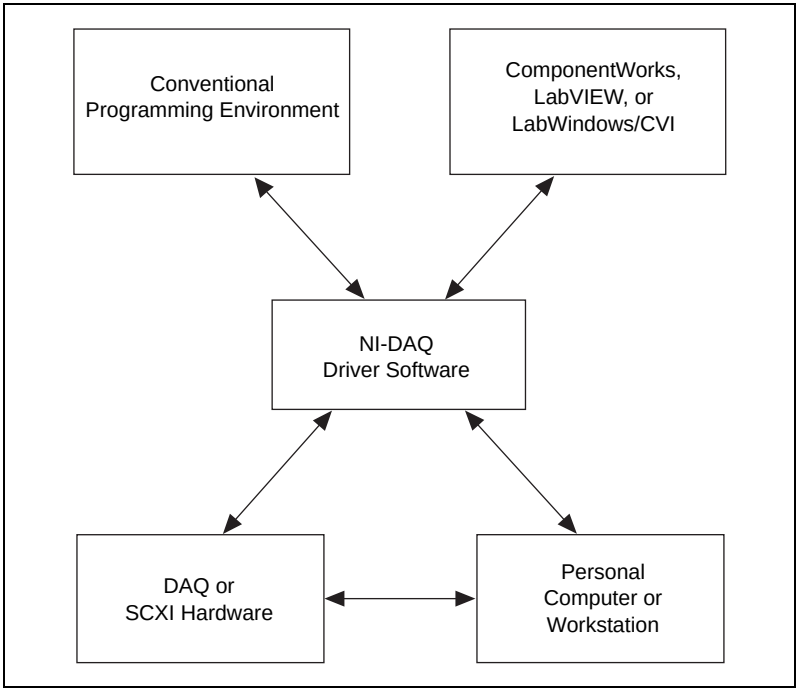



**Figure 1-1.** The Relationship between the Programming Environment, NI-DAQ, and Your Hardware

## Register-Level Programming

The final option for programming any National Instruments DAQ hardware is to write register-level software. Writing register-level programming software can be very time-consuming and inefficient, and is not recommended for most users.

Even if you are an experienced register-level programmer, consider using NI-DAQ or other National Instruments application software to program your National Instruments DAQ hardware. Using NI-DAQ, ComponentWorks, LabVIEW, or LabWindows/CVI software is easier than, and as flexible as, register-level programming, and can save weeks of development time.

# Optional Equipment

National Instruments offers a variety of products to use with your DIO board, including cables, connector blocks, and other accessories, as follows:

- Cables and cable assemblies
- Connector blocks, 50-pin screw terminals
- SCXI modules and accessories for isolating, amplifying, exciting, and multiplexing signals for relays and analog output. With SCXI you can condition and acquire up to 3,072 channels.
- Low channel count signal conditioning modules, boards, and accessories, including conditioning for strain gauges and RTDs, simultaneous sample and hold, and relays

For more information about optional equipment available from National Instruments, refer to your National Instruments catalog or call the office nearest you.

# Custom Cabling

National Instruments offers cables and accessories for you to prototype your application or to use if you frequently change board interconnections.

If you want to develop your own cable, the mating connector for the PCI-DIO-96 and PXI-6508 is a 100-position, right-angle receptacle without board locks. Recommended manufacturer part numbers for this mating connector are as follows:

- AMP Corporation (part number 749076-9)
- Honda Corporation (part number PCS-XE100LFD-HS)

The mating connector for the PCI-6503 is a 50-position, polarized ribbon socket connector with strain relief. National Instruments uses a polarized (keyed) connector to prevent inadvertent upside-down connections. Recommended manufacturer part numbers for this mating connector are as follows:

- Electronic Products Division/3M (part number 3425-7650)
- T&B/Ansley Corporation (part number 622-5041)

# Unpacking

Your DIO board is shipped in an antistatic package to prevent electrostatic damage to the board. Electrostatic discharge can damage several components on the board. To avoid such damage in handling the board, take the following precautions:

- Ground yourself via a grounding strap or by holding a grounded object.
- Touch the antistatic package to a metal part of your computer chassis before removing the board from the package.
- Remove the board from the package and inspect the board for loose components or any other sign of damage. Notify National Instruments if the board appears damaged in any way. *Do not* install a damaged board into your computer.
- *Never* touch the exposed pins of connectors.

# 2

# Installation and Configuration

This chapter describes how to install and configure your PCI-DIO-96, PXI-6508, or PCI-6503 board.

## Software Installation

Install your software before you install your DIO board. Refer to the appropriate release notes indicated below for specific instructions on the software installation sequence.

If you are using NI-DAQ, refer to your NI-DAQ release notes. Find the installation section for your operating system and follow the instructions given there.

If you are using LabVIEW, LabWindows/CVI, or other National Instruments application software packages, refer to the appropriate release notes. After you have installed your application software, refer to your NI-DAQ release notes and follow the instructions given there for your operating system and application software package.

If you are a register-level programmer, refer to Appendix B, *Register-Level Programming*, of this manual.

## Hardware Installation

Following are general installation instructions for each device. Consult your computer or chassis user manual or technical reference manual for specific instructions about installing new devices in your computer or chassis.

### Installing the PCI-DIO-96 or PCI-6503

You can install a PCI-DIO-96 or PCI-6503 in any available 5 V PCI expansion slot in your computer:

1. Turn off and unplug your computer.
2. Remove the top cover or access port to the expansion slots.

3. Remove the expansion slot cover on the back panel of the computer.
4. Touch the metal part inside your computer to discharge any static electricity that might be on your clothes or body.
5. Insert the PCI-DIO-96 or PCI-6503 in a 5 V PCI slot. It may be a tight fit, but *do not* force the device into place.
6. Screw the mounting bracket of the PCI-DIO-96 or PCI-6503 to the back panel rail of the computer.
7. Visually verify the installation.
8. Replace the top cover of your computer.
9. Plug in and turn on your computer.

## Installing the PXI-6508

You can install a PXI-6508 in any available 5 V peripheral slot in your PXI or CompactPCI chassis:

1. Turn off and unplug your PXI or CompactPCI chassis.
2. Choose an unused PXI or CompactPCI 5 V peripheral slot.
3. Remove the filler panel for the peripheral slot you have chosen.
4. Touch a metal part of your chassis to discharge any static electricity that might be on your clothes or body.
5. Insert the PXI-6508 in the selected 5 V slot. Use the injector/ejector handle to fully inject the device into place.
6. Screw the front panel of the PXI-6508 to the front panel mounting rails of the PXI or CompactPCI chassis.
7. Visually verify the installation.
8. Plug in and turn on the PXI or CompactPCI chassis.

# Board Configuration

Your DIO board is completely software configurable. The PCI-DIO-96 and PCI-6503 are fully compliant with the *PCI Local Bus Specification*, Revision 2.1, and the PXI-6508 is fully compliant with the *PXI Specification*, Revision 1.0. Therefore, all board resources are automatically allocated by the PCI system, including the base address and interrupt level. The board's base address is mapped into PCI memory space. You do not need to perform any configuration steps after the system powers up.

# 3

# Signal Connections

This chapter describes how to make input and output signal connections to your PCI-DIO-96, PXI-6508, and PCI-6503 via the board I/O connector.

**Caution** ***Connections that exceed any of the maximum ratings of input or output signals on your DIO board can damage the board and your computer. The description of each signal in this chapter includes information about maximum input ratings. National Instruments is NOT liable for any damages resulting from signal connections that exceed these maximum ratings.***

## I/O Connector (PCI-DIO-96, PXI-6508)

The I/O connector for the PCI-DIO-96 and PXI-6508 has 100 pins that you can connect to 50-pin accessories with the R1005050 cable.

### I/O Connector Pin Assignments

Figure 3-1 shows the pin assignments for the PCI-DIO-96 and PXI-6508 digital I/O connector.

| | | | |
|---|---|---|---|
| APC7 | 1 | 51 | CPC7 |
| BPC7 | 2 | 52 | DPC7 |
| APC6 | 3 | 53 | CPC6 |
| BPC6 | 4 | 54 | DPC6 |
| APC5 | 5 | 55 | CPC5 |
| BPC5 | 6 | 56 | DPC5 |
| APC4 | 7 | 57 | CPC4 |
| BPC4 | 8 | 58 | DPC4 |
| APC3 | 9 | 59 | CPC3 |
| BPC3 | 10 | 60 | DPC3 |
| APC2 | 11 | 61 | CPC2 |
| BPC2 | 12 | 62 | DPC2 |
| APC1 | 13 | 63 | CPC1 |
| BPC1 | 14 | 64 | DPC1 |
| APC0 | 15 | 65 | CPC0 |
| BPC0 | 16 | 66 | DPC0 |
| APB7 | 17 | 67 | CPB7 |
| BPB7 | 18 | 68 | DPB7 |
| APB6 | 19 | 69 | CPB6 |
| BPB6 | 20 | 70 | DPB6 |
| APB5 | 21 | 71 | CPB5 |
| BPB5 | 22 | 72 | DPB5 |
| APB4 | 23 | 73 | CPB4 |
| BPB4 | 24 | 74 | DPB4 |
| APB3 | 25 | 75 | CPB3 |
| BPB3 | 26 | 76 | DPB3 |
| APB2 | 27 | 77 | CPB2 |
| BPB2 | 28 | 78 | DPB2 |
| APB1 | 29 | 79 | CPB1 |
| BPB1 | 30 | 80 | DPB1 |
| APB0 | 31 | 81 | CPB0 |
| BPB0 | 32 | 82 | DPB0 |
| APA7 | 33 | 83 | CPA7 |
| BPA7 | 34 | 84 | DPA7 |
| APA6 | 35 | 85 | CPA6 |
| BPA6 | 36 | 86 | DPA6 |
| APA5 | 37 | 87 | CPA5 |
| BPA5 | 38 | 88 | DPA5 |
| APA4 | 39 | 89 | CPA4 |
| BPA4 | 40 | 90 | DPA4 |
| APA3 | 41 | 91 | CPA3 |
| BPA3 | 42 | 92 | DPA3 |
| APA2 | 43 | 93 | CPA2 |
| BPA2 | 44 | 94 | DPA2 |
| APA1 | 45 | 95 | CPA1 |
| BPA1 | 46 | 96 | DPA1 |
| APA0 | 47 | 97 | CPA0 |
| BPA0 | 48 | 98 | DPA0 |
| +5 V | 49 | 99 | +5 V |
| GND | 50 | 100 | GND |

**Figure 3-1.** PCI-DIO-96 and PXI-6508 Connector Pin Assignments

## Cable Assembly Connectors

The optional R1005050 cable assembly you can use with the PCI-DIO-96 or PXI-6058 is an assembly of two 50-pin cables and three connectors. Both cables are joined to a single connector on one end and to individual connectors on the free ends. The 100-pin connector that joins the two cables plugs into the I/O connector of the PCI-DIO-96 and PXI-6508. The other two connectors are 50-pin connectors, one of which is connected to pins 1 through 50 and the other connected to pins 51 through 100 of the PCI-DIO-96 and PXI-6508 connector. Figures 3-2 and 3-3 show the pin assignments for the 50-pin connectors on the cable assembly.

| | | | |
|---|---|---|---|
| APC7 | 1 | 2 | BPC7 |
| APC6 | 3 | 4 | BPC6 |
| APC5 | 5 | 6 | BPC5 |
| APC4 | 7 | 8 | BPC4 |
| APC3 | 9 | 10 | BPC3 |
| APC2 | 11 | 12 | BPC2 |
| APC1 | 13 | 14 | BPC1 |
| APC0 | 15 | 16 | BPC0 |
| APB7 | 17 | 18 | BPB7 |
| APB6 | 19 | 20 | BPB6 |
| APB5 | 21 | 22 | BPB5 |
| APB4 | 23 | 24 | BPB4 |
| APB3 | 25 | 26 | BPB3 |
| APB2 | 27 | 28 | BPB2 |
| APB1 | 29 | 30 | BPB1 |
| APB0 | 31 | 32 | BPB0 |
| APA7 | 33 | 34 | BPA7 |
| APA6 | 35 | 36 | BPA6 |
| APA5 | 37 | 38 | BPA5 |
| APA4 | 39 | 40 | BPA4 |
| APA3 | 41 | 42 | BPA3 |
| APA2 | 43 | 44 | BPA2 |
| APA1 | 45 | 46 | BPA1 |
| APA0 | 47 | 48 | BPA0 |
| +5 V | 49 | 50 | GND |

**Figure 3-2.** Cable-Assembly Connector Pinout for Pins 1 through 50 with the R1005050 Ribbon Cable (PCI-DIO-96 and PXI-6508)

| | | | |
|---|---|---|---|
| CPC7 | 51 | 52 | DPC7 |
| CPC6 | 53 | 54 | DPC6 |
| CPC5 | 55 | 56 | DPC5 |
| CPC4 | 57 | 58 | DPC4 |
| CPC3 | 59 | 60 | DPC3 |
| CPC2 | 61 | 62 | DPC2 |
| CPC1 | 63 | 64 | DPC1 |
| CPC0 | 65 | 66 | DPC0 |
| CPB7 | 67 | 68 | DPB7 |
| CPB6 | 69 | 70 | DPB6 |
| CPB5 | 71 | 72 | DPB5 |
| CPB4 | 73 | 74 | DPB4 |
| CPB3 | 75 | 76 | BPB3 |
| CPB2 | 77 | 78 | DPB2 |
| CPB1 | 79 | 80 | DPB1 |
| CPB0 | 81 | 82 | DPB0 |
| CPA7 | 83 | 84 | DPA7 |
| CPA6 | 85 | 86 | DPA6 |
| CPA5 | 87 | 88 | DPA5 |
| CPA4 | 89 | 90 | DPA4 |
| CPA3 | 91 | 92 | DPA3 |
| CPA2 | 93 | 94 | DPA2 |
| CPA1 | 95 | 96 | DPA1 |
| CPA0 | 97 | 98 | DPA0 |
| +5 V | 99 | 100 | GND |

**Figure 3-3.** Cable-Assembly Connector Pinout for Pins 51 through 100 with the R1005050 Ribbon Cable (PCI-DIO-96 and PXI-6508)

## I/O Connector Signal Descriptions

Table 3-1 lists the signal descriptions for the PCI-DIO-96 and PXI-6508 I/O connector pins.

**Table 3-1.** Signal Descriptions for PCI-DIO-96 and PXI-6508 I/O Connectors

| Pin | Signal Name | Alternate Port ID* | Description |
|---|---|---|---|
| 1, 3, 5, 7, 9, 11, 13, 15 | APC<7..0> | 2 | Bidirectional data lines for port C of PPI A—APC7 is the MSB, APC0 the LSB. |
| 2, 4, 6, 8, 10, 12, 14, 16 | BPC<7..0> | 5 | Bidirectional data lines for port C of PPI B—BPC7 is the MSB, BPC0 the LSB. |
| 17, 19, 21, 23, 25, 27, 29, 31 | APB<7..0> | 1 | Bidirectional data lines for port B of PPI A—APB7 is the MSB, APB0 the LSB. |
| 18, 20, 22, 24, 26, 28, 30, 32 | BPB<7..0> | 4 | Bidirectional data lines for port B of PPI B—BPB7 is the MSB, BPB0 the LSB. |
| 33, 35, 37, 39, 41, 43, 45, 47 | APA<7..0> | 0 | Bidirectional data lines for port A of PPI A—APA7 is the MSB, APA0 the LSB. |
| 34, 36, 38, 40, 42, 44, 46, 48 | BPA<7..0> | 3 | Bidirectional data lines for port A of PPI B—BPA7 is the MSB, BPA0 the LSB. |
| 49, 99 | +5 V supply | – | +5 Volts—These pins are fused for up to 1 A total of +4.65 to +5.25 V. |
| 50, 100 | GND | – | Ground—These pins are connected to the computer ground signal. |
| 51, 53, 55, 57, 59, 61, 63, 65 | CPC<7..0> | 8 | Bidirectional data lines for port C of PPI C—CPC7 is the MSB, CPC0 the LSB. |
| 52, 54, 56, 58, 60, 62, 64, 66 | DPC<7..0> | 11 | Bidirectional data lines for port C of PPI D—DPC7 is the MSB, DPC0 the LSB. |

**Table 3-1.** Signal Descriptions for PCI-DIO-96 and PXI-6508 I/O Connectors (Continued)

| Pin | Signal Name | Alternate Port ID* | Description |
|---|---|---|---|
| 67, 69, 71, 73, 75, 77, 79, 81 | CPB<7..0> | 7 | Bidirectional data lines for port B of PPI C—CPB7 is the MSB, CPB0 the LSB. |
| 68, 70, 72, 74, 76, 78, 80, 82 | DPB<7..0> | 10 | Bidirectional data lines for port B of PPI D—DPB7 is the MSB, DPB0 the LSB. |
| 83, 85, 87, 89, 91, 93, 95, 97 | CPA<7..0> | 6 | Bidirectional data lines for port A of PPI C—CPA7 is the MSB, CPA0 the LSB. |
| 84, 86, 88, 90, 92, 94, 96, 98 | DPA<7..0> | 9 | Bidirectional data lines for port A of PPI D—DPA7 is the MSB, DPA0 the LSB. |

* This document refers to the ports as A, B, and C and the PPIs (82C55As) as A, B, C, and D. NI-DAQ and LabVIEW documentation use numbers to identify each port and PPI. For example, this manual uses PPI A port A to refer to port A of the 82C55A identified as PPI A. NI-DAQ, LabWindows/CVI, LabVIEW, or other application software documentation, however, refer to this port as *0*. The Alternate Port ID column shows the correlation between the different port names.

# I/O Connector (PCI-6503)

The PCI-6503 has 50 pins that you can connect to 50-pin accessories with the NB1 cable.

## PCI-6503 I/O Connector Pin Descriptions

Figure 3-4 shows the pin assignments for the PCI-6503 digital I/O connector using the NB1 ribbon cable.

| | | | |
|---|---|---|---|
| PC7 | 1 | 2 | GND |
| PC6 | 3 | 4 | GND |
| PC5 | 5 | 6 | GND |
| PC4 | 7 | 8 | GND |
| PC3 | 9 | 10 | GND |
| PC2 | 11 | 12 | GND |
| PC1 | 13 | 14 | GND |
| PC0 | 15 | 16 | GND |
| PB7 | 17 | 18 | GND |
| PB6 | 19 | 20 | GND |
| PB5 | 21 | 22 | GND |
| PB4 | 23 | 24 | GND |
| PB3 | 25 | 26 | GND |
| PB2 | 27 | 28 | GND |
| PB1 | 29 | 30 | GND |
| PB0 | 31 | 32 | GND |
| PA7 | 33 | 34 | GND |
| PA6 | 35 | 36 | GND |
| PA5 | 37 | 38 | GND |
| PA4 | 39 | 40 | GND |
| PA3 | 41 | 42 | GND |
| PA2 | 43 | 44 | GND |
| PA1 | 45 | 46 | GND |
| PA0 | 47 | 48 | GND |
| +5 V | 49 | 50 | GND |

**Figure 3-4.** PCI-6503 I/O Connector Pin Assignments

Table 3-2 describes the PCI-6503 signals.

**Table 3-2.** PCI-6503 Signal Descriptions

| Pin | Signal Name | Alternate Port ID* | Description |
|---|---|---|---|
| 1, 3, 5, 7, 9, 11, 13, 15 | PC<7..0> | 2 | Port C—Bidirectional data lines for port C. PC7 is the MSB, PC0 the LSB. |
| 17, 19, 21, 23, 25, 27, 29, 31 | PB<7..0> | 1 | Port B—Bidirectional data lines for port B. PB7 is the MSB, PB0 the LSB. |
| 33, 35, 37, 39, 41, 43, 45, 47 | PA<7..0> | 0 | Port A—Bidirectional data lines for port B. PA7 is the MSB, PA0 the LSB. |
| 49 | +5 V | — | +5 Volts—This pin is fused for up to 1 A at +4.65 to 5.25 V. |
| All even-numbered pins | GND | — | Ground—These signals are connected to the computer ground reference. |

*This document refers to the 82C55 ports as A, B, and C. NI-DAQ and LabVIEW documentation use numbers to identify ports. For example, this manual uses port A to refer to the first port of the 82C55A. NI-DAQ, LabWindows/CVI, LabVIEW, or other application software documentation, however, refer to this port as *0*. The Alternate Port ID column shows the correlation between the different port names.

# Port C Pin Assignments

The signals assigned to port C depend on how the 82C55A is configured. In mode 0, or no handshaking configuration, port C is configured as two 4-bit I/O ports. In modes 1 and 2, or handshaking configuration, port C is used for status and handshaking signals with any leftover lines available for general-purpose I/O. Table 3-3 summarizes the port C signal assignments for each configuration. You can also use ports A and B in different modes; the table does not show every possible combination. Consult Appendix B, *Register-Level Programming*, for register-level programming information.

**Note** ***Table 3-3 shows both the port C signal assignments and the terminology correlation between different documentation sources. The 82C55A terminology refers to the different 82C55A configurations as modes whereas NI-DAQ, ComponentWorks, LabWindows/CVI, and LabVIEW documentation refers to them as handshaking and no handshaking. On the PCI-DIO-96 and PXI-6508, these signal assignments are the same for all four 82C55A PPIs. Refer to Table 3-1 for more information.***

**Table 3-3.** Port C Signal Assignments

| **Configuration Terminology** | | **Signal Assignments** | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **82C55A/ *PCI-DIO-96/ PXI-6508/ PCI-6503 User Manual*** | **National Instruments Software** | **APC7, BPC7, CPC7, or DPC7** | **APC6, BPC6, CPC6, or DPC6** | **APC5, BPC5, CPC5, or DPC5** | **APC4, BPC4, CPC4, or DPC4** | **APC3, BPC3, CPC3, or DPC3** | **APC2, BPC2, CPC2, or DPC2** | **APC1, BPC1, CPC1, or DPC1** | **APC0, BPC0, CPC0, or DPC0** |
| Mode 0 (Basic I/O) | No Handshaking | I/O | I/O | I/O | I/O | I/O | I/O | I/O | I/O |
| Mode 1 (Strobed Input) | Handshaking | I/O | I/O | $IBF_A$ | $STB_A$* | $INTR_A$ | $STB_B$* | $IBFB_B$ | $INTR_B$ |
| Mode 1 (Strobed Output) | Handshaking | $OBF_A$* | $ACK_A$* | I/O | I/O | $INTR_A$ | $ACK_B$* | $OBF_B$* | $INTR_B$ |
| Mode 2 (Bidirectional Bus) | Handshaking | $OBF_A$* | $ACK_A$* | $IBF_A$ | $STB_A$* | $INTR_A$ | I/O | I/O | I/O |

*Indicates that the signal is active low.
Subscripts A and B denote port A or port B handshaking signals.

# Digital I/O Signal Connections

Pins 1 through 48 and, on the PCI-DIO-96 and PXI-6508, pins 51 through 98 of the I/O connector are digital I/O signal pins. The following specifications and ratings apply to the digital I/O lines. The maximum input logic high and output logic high voltages assume a $V_{cc}$ supply voltage of 5.0 V.

The absolute maximum voltage rating is –0.5 to +5.5 V with respect to GND.

Digital input specifications (referenced to GND):

| | | |
|---|---|---|
| Input logic high voltage | 2.2 V min | 5.3 V max |
| Input logic low voltage | –0.3 V min | 0.8 V max |
| Input high current ($V_{in}$ = 5 V, resistors set to pull-up[1]) | — | 10 µA max[2] |
| Input high current ($V_{in}$ = 5 V, resistors set to pull-down[1]) | — | 75 µA max |
| Input logic low current ($V_{in}$ = 0 V, resistors set to pull-up[1]) | — | –75 µA max |
| Input logic low current ($V_{in}$ = 0 V, resistors set to pull-down[1]) | — | –10 µA max[2] |

Digital output specifications (referenced to GND):

| | | |
|---|---|---|
| Output logic high voltage ($I_{ol}$ = –2.5 mA) | 3.7 V min | 5.0 V max |
| Output logic high voltage ($I_{oh}$ = –4 mA) | 2.7 V min | 5.0 V max |
| Output logic low voltage ($I_{ol}$ = 2.5 mA) | 0 V min | 0.4 V |
| Output logic low voltage ($I_{ol}$ = 4 mA) | 0 V min | 0.5 V |

[1] The PCI-DIO-96 bias resistors are always set to pull-up. On the PXI-6508 and PCI-6503, you can use jumper W1 to select pull-up or pull-down.

[2] Exception: Lines PC3 and PC0 are 20 µA.

Figure 3-5 depicts signal connections for three typical digital I/O applications.

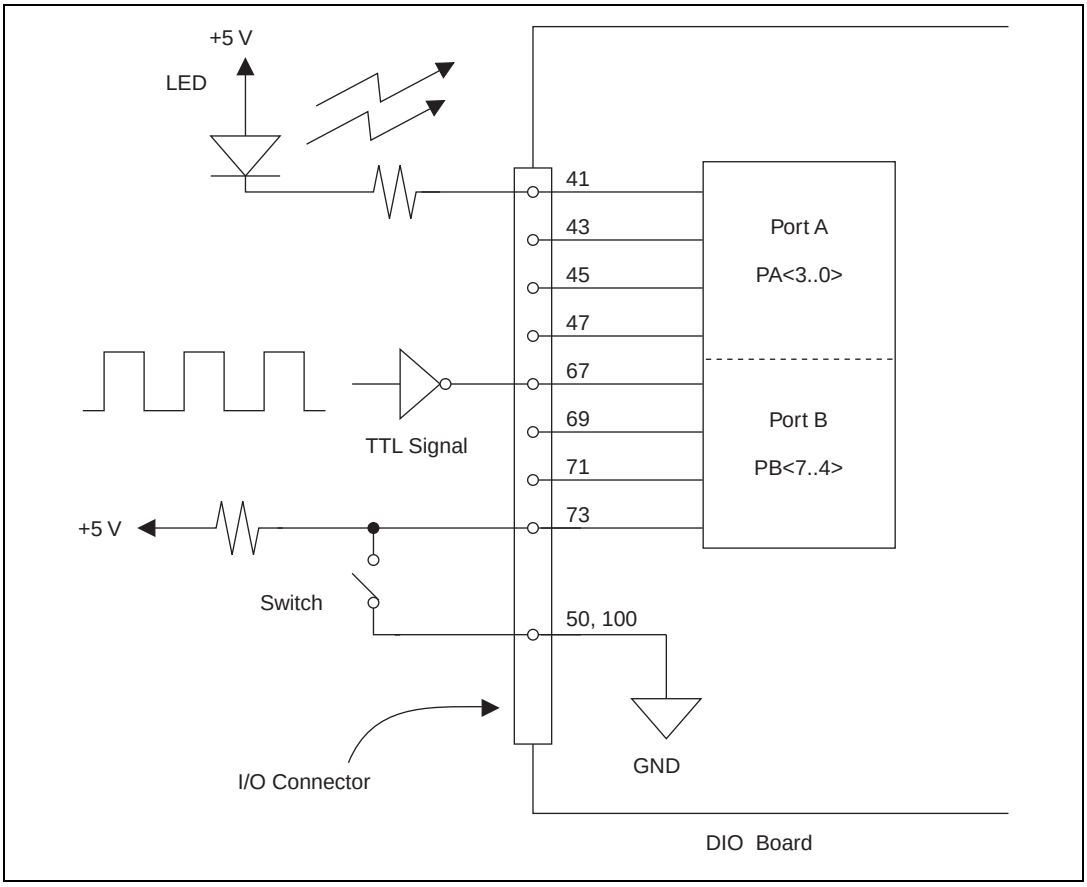



**Figure 3-5.** Digital I/O Connections Block Diagram

In Figure 3-5, port A of one PPI is configured for digital output, and port B is configured for digital input. Digital input applications include receiving TTL signals and sensing external device states such as the state of the switch in Figure 3-5. Digital output applications include sending TTL signals and driving external devices such as the LED shown in Figure 3-5.

# Power Connections

Pin 49 and, on the PCI-DIO-96 and PXI-6508, pin 99 of the I/O connector supply +5 V from the computer's power supply via a self-resetting fuse. The fuse will reset automatically within a few seconds after the overcurrent condition is removed. These pins are referenced to GND and can be used to power external digital circuitry.

- Power rating 1 A at +4.65 to +5.25 V

**Caution** ***Under no circumstances should you connect these +5 V power pins directly to ground or to any other voltage source on your DIO board or any other device. Doing so can damage your DIO board and the computer. National Instruments is NOT liable for damage resulting from such a connection.***

# Digital I/O Power-up State Selection

The PCI-DIO-96, PXI-6508, and PCI-6503 contain bias resistors that control the state of the digital I/O lines at power up. At power up, each digital I/O line is configured as an input, pulled either high or low by a 100 kΩ bias resistor.

On the PCI-DIO-96, all of the 100 kΩ bias resistors pull up. Therefore, the default power-up state of each line on the PCI-DIO-96 is high.

On the PXI-6508 and PCI-6503, you can select the direction of the 100 kΩ bias resistors. Set jumper W1 to high to configure the resistors as pull-up resistors. Set jumper W1 to low to configure the resistors as pull-down resistors.

You can change individual lines from pulled up to pulled down—or, on the PXI-6508 and PCI-6503, from pulled down to pulled up—by adding your own external resistors. This section describes the procedure.

## High DIO Power-up State

If you select the pulled-high mode, each DIO line will be pulled to $V_{cc}$ (approximately +5 VDC) with a 100 kΩ resistor. If you want to pull a specific line low, connect between that line and ground a pull-down resistor ($R_L$) whose value will give you a maximum of 0.4 VDC. The DIO lines provide a maximum of 2.5 mA at 3.7 V in the high state. Using the largest possible resistor ensures that you do not use more current than necessary to perform the pull-down task.

However, make sure the resistor's value is not so large that leakage current from the DIO line along with the current from the 100 kΩ pull-up resistor drives the voltage at the resistor above a TTL low level of 0.4 VDC.

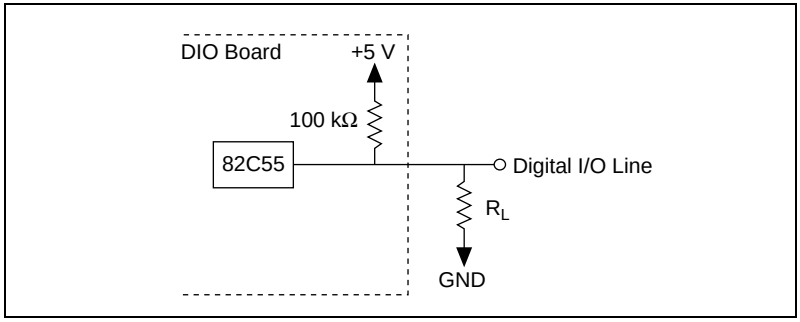



**Figure 3-6.** DIO Channel Configured for High DIO Power-up State with External Load

Example:

By default, all DIO lines are pulled high at power up. To pull one channel low, follow these steps:

1. Install a load ($R_L$). Remember that the smaller the resistance, the greater the current consumption and the lower the voltage.
2. Using the following formula, calculate the largest possible load to maintain a logic low level of 0.4 V and supply the maximum driving current:

   $V = I * R_L \Rightarrow R_L = V/I$, where:

   $V = 0.4$ V ; Voltage across $R_L$

   $I = 46\ \mu A + 10\ \mu A$ ; 4.6 V across the 100 kΩ pull-up resistor and 10 µA maximum leakage current (except lines PC0 and PC3)

   Therefore:

   $R_L = 7.1$ kΩ ; 0.4 V/56 µA

This resistor value, 7.1 kΩ, provides a maximum of 0.4 V on the DIO line at power up. You can substitute smaller resistor values to lower the voltage or to provide a margin for $V_{cc}$ variations and other factors. However, smaller values will draw more current, leaving less drive current for other circuitry connected to this line. The 7.1 kΩ resistor reduces the amount of logic high source current by 0.4 mA with a 2.8 V output.

The maximum leakage current on most lines is 10 µA. The maximum leakage current on the PC(0) and PC(3) lines is 20 µA.

## Low DIO Power-up State (PXI-6508, PCI-6503 Only)

If you select pulled-low mode, each DIO line will be pulled to GND (0 VDC) using a 100 kΩ resistor. If you want to pull a specific line high, connect a pull-up resistor that will give you a minimum of 2.8 VDC. The DIO lines are capable of sinking a maximum of 2.5 mA at 0.4 V in the low state. Using the largest possible resistance value ensures that you do not use more current than necessary to perform the pull-up task.

Also, make sure the pull-up resistor value is not so large that leakage current from the DIO line along with the current from the 100 kΩ pull-down resistor brings the voltage at the resistor below a TTL high level of 2.8 VDC.

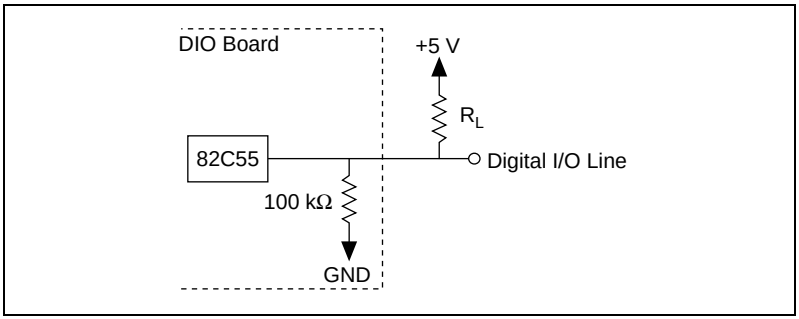



**Figure 3-7.** DIO Channel Configured for Low DIO Power-up State with External Load

Example:

Set jumper W1 to low, which means all DIO lines are pulled low at power up. If you want to pull one channel high, follow these steps:

1. Install a load ($R_L$). Remember that the smaller the resistance, the greater the current consumption and the higher the voltage.
2. Using the following formula, calculate the largest possible load to maintain a logic high level of 2.8 V and supply the maximum sink current:

   $V = I * R_L \Rightarrow R_L = V/I$, where:

   $V = 2.2$ V ; Voltage across $R_L$

   $I = 28\ \mu A + 10\ \mu A$ ; 2.8 V across the 100 kΩ pull-up resistor and 10 μA maximum leakage current (except lines PC0 and PC3)

   Therefore:

   $R_L = 5.7$ kΩ ; 2.2 V/38 μA

This resistor value, 5.7 kΩ, provides a maximum of 2.8 V on the DIO line at power up. You can substitute smaller resistor values to lower the voltage drop or to provide a margin for $V_{CC}$ variations and other factors. However, smaller values will draw more current, leaving less sink current for other circuitry connected to this line. The 5.7 kΩ resistor will reduce the amount of a logic low sink current by 0.8 mA with a 0.4 V output.

# Timing Specifications

This section lists the timing specifications for handshaking with your DIO board. The handshaking lines STB* and IBF synchronize input transfers. The handshaking lines OBF* and ACK* synchronize output transfers.

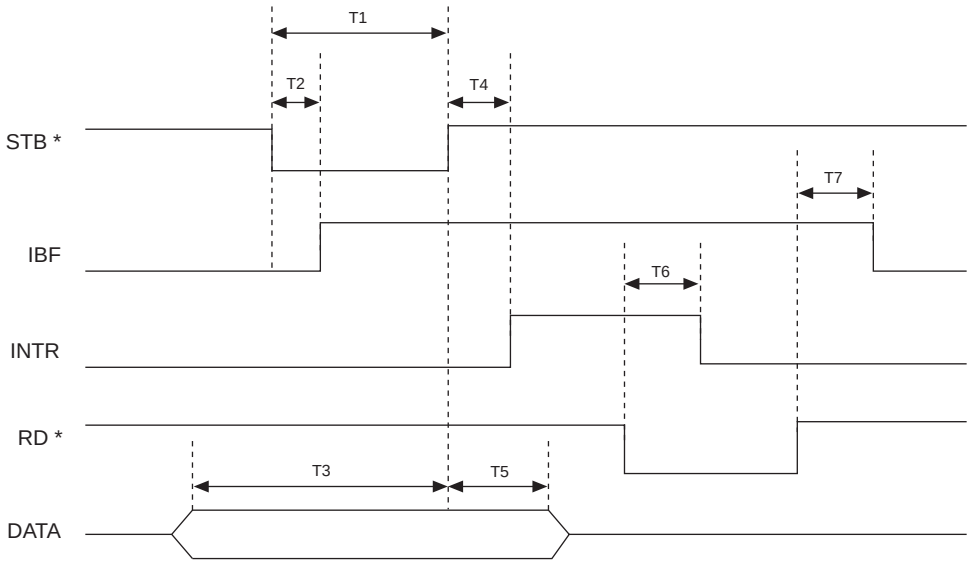
Table 3-4 describes signals appearing in the handshaking diagrams.

**Table 3-4.** Signal Names Used in Timing Diagrams

| Name | Type | Description |
|---|---|---|
| STB* | Input | Strobe Input—A low signal on this handshaking line loads data into the input latch. |
| IBF | Output | Input Buffer Full—A high signal on this handshaking line indicates that data has been loaded into the input latch. A low signal indicates the board is ready for more data. This is an input acknowledge signal. |
| ACK* | Input | Acknowledge Input—A low signal on this handshaking line indicates that the data written to the port has been accepted. This signal is a response from the external device indicating that it has received the data from your DIO board. |
| OBF* | Output | Output Buffer Full—A low signal on this handshaking line indicates that data has been written to the port. |
| INTR | Output | Interrupt Request—This signal becomes high when the 82C55A requests service during a data transfer. The appropriate interrupt enable bits must be set to generate this signal. |
| RD* | Internal | Read—This signal is the read signal generated from the control lines of the computer I/O expansion bus. |
| WR* | Internal | Write—This signal is the write signal generated from the control lines of the computer I/O expansion bus. |
| DATA | Bidirectional | Data Lines at the Specified Port—For output mode, this signal indicates the availability of data on the data line. For input mode, this signal indicates when the data on the data lines should be valid. |

## Mode 1 Input Timing

The timing specifications for an input transfer in mode 1 are as follows:



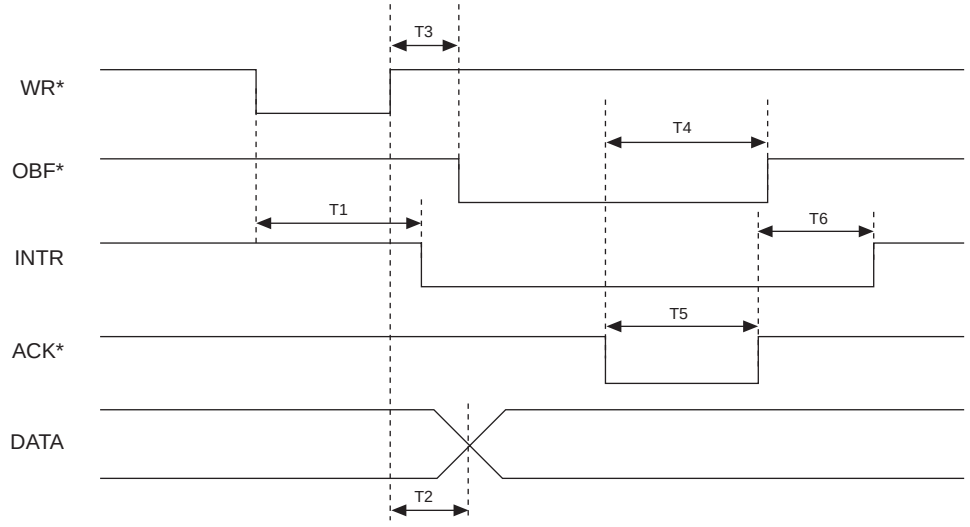

| Name | Description | Minimum | Maximum |
|---|---|---|---|
| T1 | STB* Pulse Width | 100 | — |
| T2 | STB* = 0 to IBF = 1 | — | 150 |
| T3 | Data before STB* = 1 | 20 | — |
| T4 | STB* = 1 to INTR = 1 | — | 150 |
| T5 | Data after STB* = 1 | 50 | — |
| T6 | RD* = 0 to INTR = 0 | — | 200 |
| T7 | RD* = 1 to IBF = 0 | — | 150 |

All timing values are in nanoseconds.

**Figure 3-8.** Timing Specifications for Mode 1 Input Transfer

## Mode 1 Output Timing

The timing specifications for an output transfer in mode 1 are as follows:



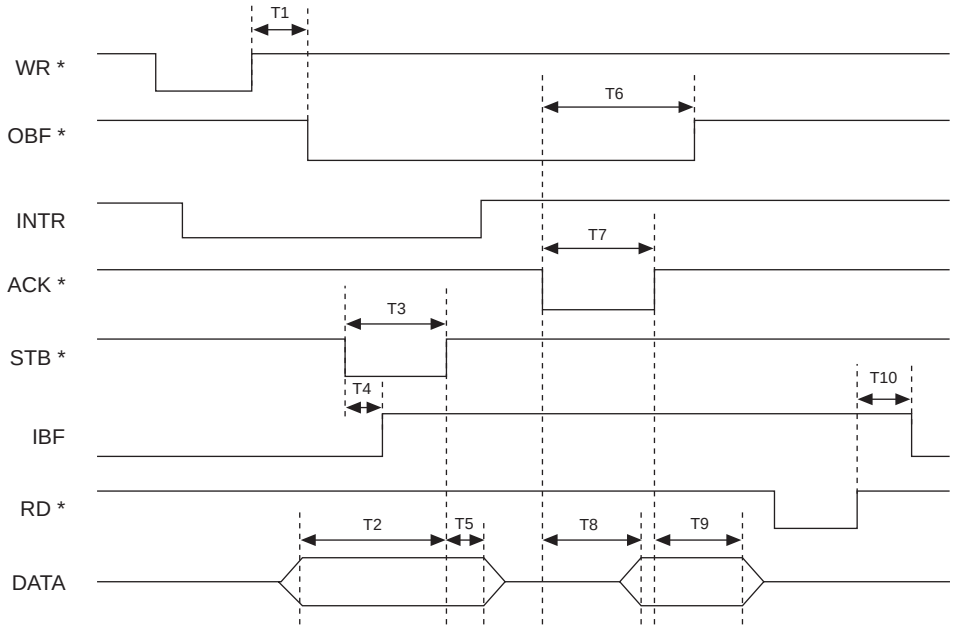

| Name | Description | Minimum | Maximum |
|---|---|---|---|
| T1 | WR* = 0 to INTR = 0 | — | 250 |
| T2 | WR* = 1 to Output | — | 200 |
| T3 | WR* = 1 to OBF* = 0 | — | 150 |
| T4 | ACK* = 0 to OBF* = 1 | — | 150 |
| T5 | ACK* Pulse Width | 100 | — |
| T6 | ACK* = 1 to INTR = 1 | — | 150 |

All timing values are in nanoseconds.

**Figure 3-9.** Timing Specifications for Mode 1 Output Transfer

## Mode 2 Bidirectional Timing

The timing specifications for bidirectional transfers in mode 2 are as follows:




| Name | Description | Minimum | Maximum |
|---|---|---|---|
| T1 | WR* = 1 to OBF* = 0 | — | 150 |
| T2 | Data before STB* = 1 | 20 | — |
| T3 | STB* Pulse Width | 100 | — |
| T4 | STB* = 0 to IBF = 1 | — | 150 |
| T5 | Data after STB* = 1 | 50 | — |
| T6 | ACK* = 0 to OBF* = 1 | — | 150 |
| T7 | ACK* Pulse Width | 100 | — |
| T8 | ACK* = 0 to Output | — | 150 |
| T9 | ACK* = 1 to Output Float | 20 | 250 |
| T10 | RD* = 1 to IBF = 0 | — | 150 |

All timing values are in nanoseconds.

**Figure 3-10.** Timing Specifications for Mode 2 Bidirectional Transfer

# 4

# Theory of Operation

This chapter contains a functional overview of the PCI-DIO-96, PXI-6508, and PCI-6503 and explains the operation of each functional unit.

## Functional Overview

The block diagram in Figure 4-1 illustrates the key functional components of your DIO board.

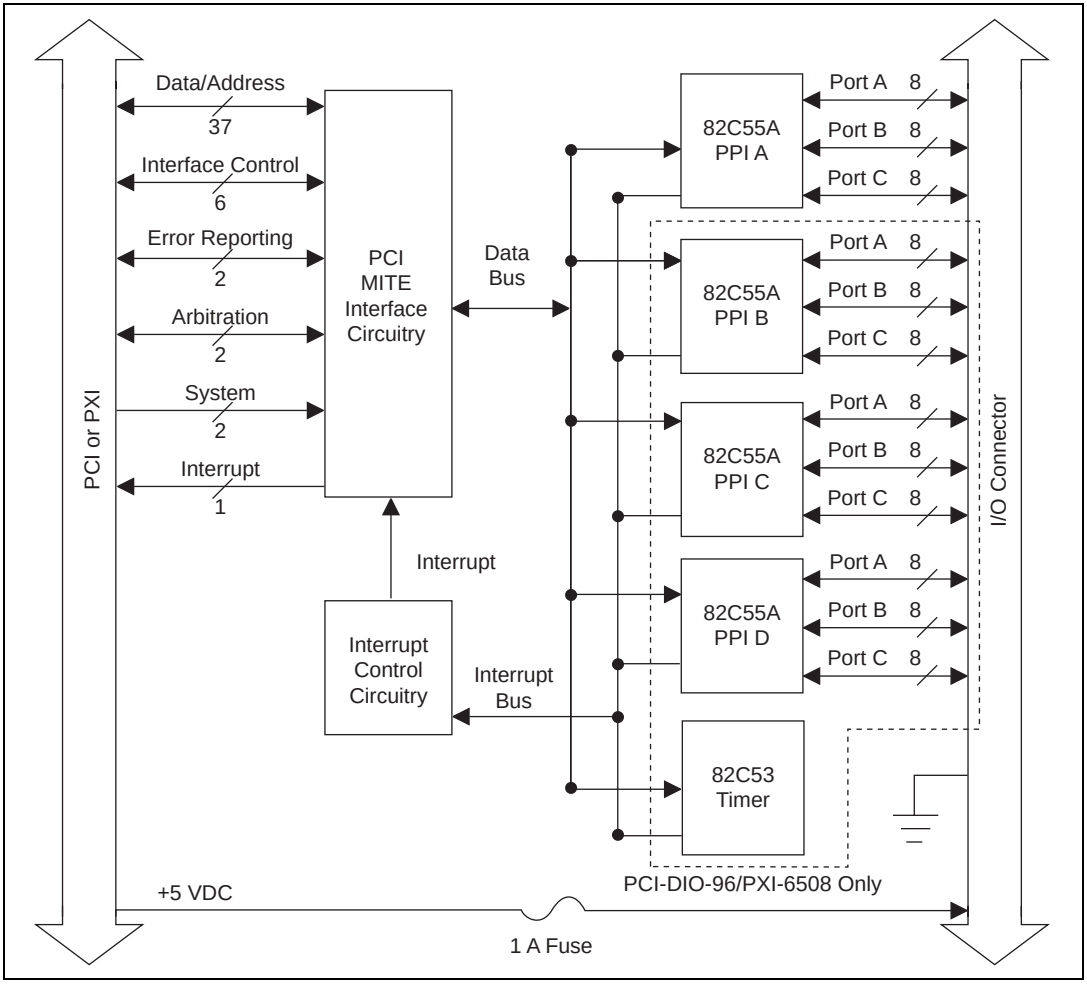



**Figure 4-1.** Block Diagram

## PCI Interface Circuitry

Your DIO board uses the PCI MITE ASIC to communicate with the PCI bus. The PCI MITE ASIC was designed by National Instruments specifically for data acquisition. The PCI MITE is fully compliant with *PCI Local Bus Specification 2.1*.

The base memory address and interrupt level for the board are stored inside the PCI MITE at power on. You do not need to set any switches or jumpers.

## 82C55A Programmable Peripheral Interface

The 82C55A PPI chip is the heart of your DIO board. The PCI-DIO-96 and PXI-6508 contain four PPIs. The PCI-6503 contains one PPI. Each of these chips has 24 programmable I/O pins that represent three 8-bit ports: PA, PB, and PC. Each port can be programmed as an input or output port. The 82C55A has three modes of operation: simple I/O (mode 0), strobed I/O (mode 1), and bidirectional I/O (mode 2). In modes 1 and 2, the three ports are divided into two groups: group A and group B. Each group has eight data bits, plus control and status bits from port C (PC). Modes 1 and 2 use handshaking signals from the computer to synchronize data transfers. Refer to Appendix B, *Register-Level Programming*, or to Appendix C, *MSM82C55A Data Sheet*, for more detailed information.

## 82C53 Programmable Interval Timer (PCI-DIO-96, PXI-6508 Only)

The PCI-DIO-96 and PXI-6508 contain an 82C53 programmable interval timer for use by register-level programmers only. The 82C53 programmable interval timer can generate timed interrupt requests to your computer. The 82C53 has three 16-bit counters, which can each be used in one of six different modes. The PCI-DIO-96 and PXI-6508 can use two of the counters to generate interrupt requests; the third counter is not used and is not accessible. Refer to Appendix B, *Register-Level Programming*, or to Appendix D, *MSM82C53 Data Sheet*, for more detailed information.

## Interrupt Control Circuitry

Two software-controlled registers determine which devices, if any, generate interrupts. Each of the 82C55A devices has two interrupt lines, PC3 and PC0, connected to the interrupt circuitry. On the PCI-DIO-96 and PXI-6508, the 82C53 device has two of its three counter outputs connected to the interrupt circuitry. Any of these 10 signals can interrupt the computer if the interrupt circuitry is enabled and the corresponding enable bit is set. See Appendix B, *Register-Level Programming*, for more information. Normally, the handshaking circuitry controls PC3 and PC0 of the 82C55A devices; however, you can configure either of these two lines for input and then use them as external interrupts. An interrupt occurs on the signal line low-to-high transition.

Refer to Appendix B, *Register-Level Programming*, Appendix C, *MSM82C55A Data Sheet*, or Appendix D, *MSM82C53 Data Sheet*, for more detailed information concerning interrupts.

The block diagram in Figure 4-2 illustrates the interrupt control circuitry.

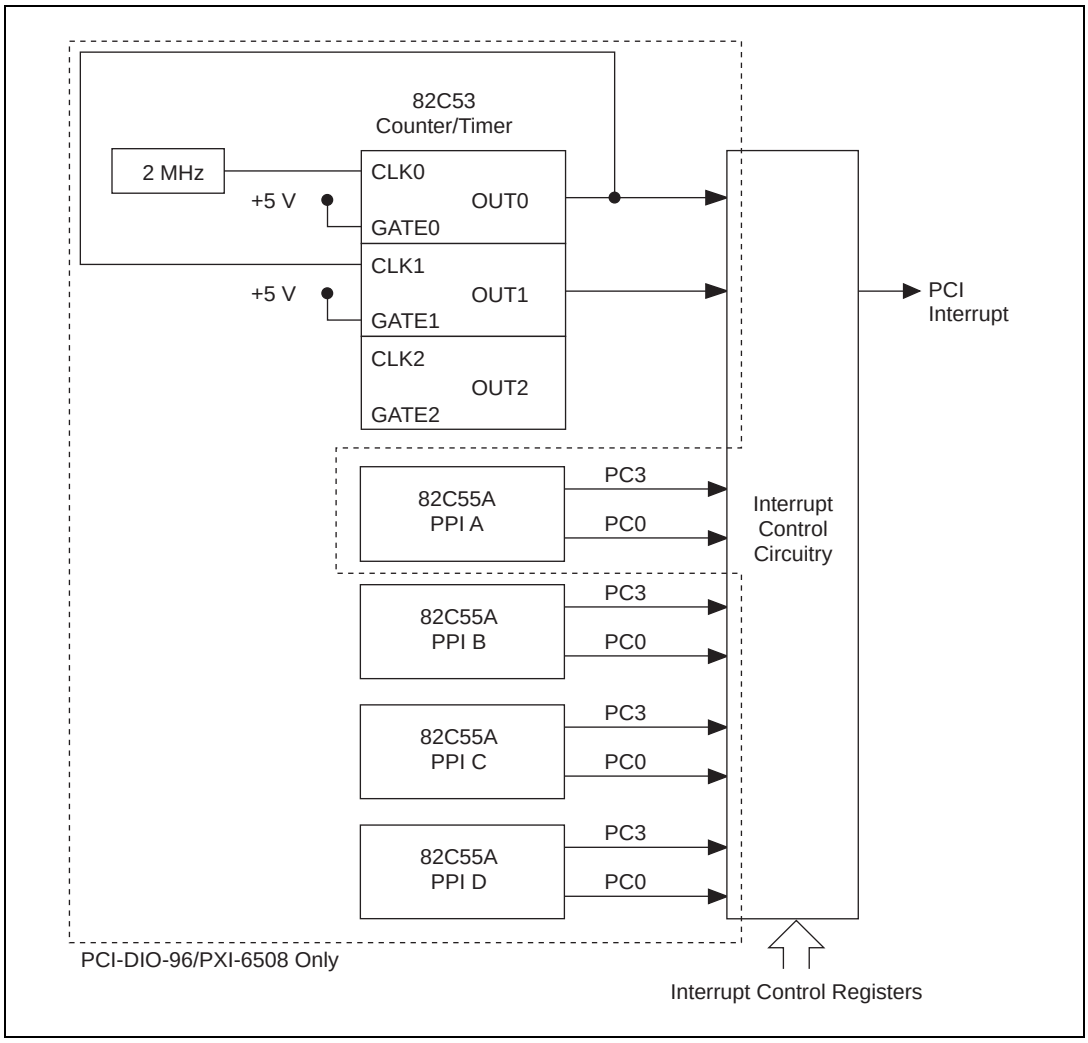



**Figure 4-2.** Interrupt Control Circuitry Block Diagram

# A

# Specifications

This appendix lists the specifications for the PCI-DIO-96, PXI-6508, and PCI-6503. These specifications are typical at 25° C unless otherwise noted.

## Digital I/O

Number of channels

- PCI-DIO-96 and PXI-6508 ..... 96 I/O
- PCI-6503 ..... 24 I/O

Compatibility ..... TTL

Power on state

- PCI-DIO-96 ..... Inputs (high-Z), pulled up through 100 kΩ
- PXI-6508, PCI-6503 ..... Inputs (high-Z), pulled up or down through 100 kΩ (jumper selectable)

Handshaking..... Input, output, or bidirectional

Data transfers ..... Interrupts, programmed I/O

## Digital Logic Levels

### Input Signals

The maximum input logic high and output logic high voltages assume a $V_{cc}$ supply voltage of 5.0 V. Given a $V_{cc}$ supply voltage of 5.0 V, the absolute maximum voltage rating for each I/O line is –0.5 V to 5.5V with respect to GND.

| Level | Min | Max |
|---|---|---|
| Input logic high voltage | 2.2 V | 5.3 V |
| Input logic low voltage | –0.3 V | 0.8 V |
| Input high current ($V_{in}$ = 5 V, resistors set to pull-up[1]) | — | 10 µA[2] |
| Input high current ($V_{in}$ = 5 V, resistors set to pull-down[1]) | — | 75 µA |
| Input logic low current ($V_{in}$ = 0 V, resistors set to pull-up[1]) | — | –75 µA |
| Input logic low current ($V_{in}$ = 0 V, resistors set to pull-down[1]) | — | –10 µA[2] |

[1] The PCI-DIO-96 bias resistors are always set to pull-up. On the PXI-6508 and PCI-6503, you can use jumper W1 to select pull-up or pull-down.

[2] Exception: Lines PC3 and PC0 are 20 µA.

## Output Signals

Pin 49 (at +5 V) .....1.0 A max

| Level | Min | Max |
|---|---|---|
| Output logic high voltage ($I_{ol}$ = -2.5 mA) | 3.7 V | 5.0 V |
| Output logic high voltage ($I_{oh}$ = -4 mA) | 2.7 V | 5.0 V |
| Output logic low voltage ($I_{ol}$ = 2.5 mA) | 0 V | 0.4 V |
| Output logic low voltage ($I_{ol}$ = 4 mA) | 0 V | 0.5 V |

## Transfer Rates

Max with NI-DAQ software ..... 50 kbytes/s

Constant sustainable rate (typ)..... 1 to 10 kbytes/s

Transfer rates are a function of the speed with which your program reads data from or writes data to the board, and therefore vary with your system, software, and application. The following primary factors control your DIO board transfer rates:

- Computer system performance
- Programming environment (register-level programming or NI-DAQ)
- Programming language and code efficiency
- Execution mode (foreground or background, with background execution typically using interrupts)
- Other operations in progress
- Application

For example, you can obtain higher transfer rates in a handshaking or data-transfer application, requiring an average rate, than in a pattern generation, data acquisition, or waveform generation application, requiring a constant sustainable rate.

The maximum rate shown was obtained using a 233 MHz Pentium computer running NI-DAQ and LabWindows/CVI software, with interrupt-based execution, and with no other high-speed operations in progress.

## Bus Interface

Type ..... Slave

## Power Requirement

Power consumption..... 400 mA at +5 VDC (±5%)

Power available at I/O connector..... +4.65 to +5.25 V fused at 1 A

## Physical

Dimensions

PCI-DIO-96 ..... 13.7 × 10.7 cm (5.4 × 4.2 in.)

PXI-6508 ..... 17.5 × 10.7 cm (6.9 × 4.2 in.)

PCI-6503..... 12.2 × 9.5 cm (4.8 × 3.7 in.)

I/O connector

PCI-DIO-96 and PXI-6508..... 100-pin female 0.050 series D-type

PCI-6503..... 50-pin male ribbon-cable connector

## Environment

Operating temperature ..... 0° to 55° C

Storage temperature ..... –20° to 70° C

Relative humidity ..... 5% to 90% noncondensing

Functional shock (PXI-6508) ..... MIL-T-28800 E Class 3 (per Section 4.5.5.4.1); half-sine shock pulse, 11 ms duration, 30 g peak, 30 shocks per face

Operational random vibration (PXI-6508)..... 5 to 500 Hz, 0.31 grms, 3 axes

Nonoperational random vibration (PXI-6508)..... 5 to 500 Hz, 2.5 grms, 3 axes

**Note** ***Random vibration profiles were developed in accordance with MIL-T-28800E and MIL-STD-810E Method 514. Test levels exceed those recommended in MIL-STD-810E for Category 1 (Basic Transportation, Figures 514.4-1 through 514.4-3).***

# B

# Register-Level Programming

This appendix describes in detail the address and function of each PCI-DIO-96, PXI-6508, and PCI-6503 register, contains instructions on how to operate DIO board circuitry, and provides examples of the programming steps necessary to execute an operation.

☞ **Note** ***If you plan to use a programming software package such as ComponentWorks, LabVIEW, LabWindows/CVI, or NI-DAQ with your DIO board, you need not read this appendix.***

## Register Map and Description

This section describes in detail the address and function of each PCI-DIO-96, PXI-6508, and PCI-6503 register.

### Introduction

The three 8-bit ports of the 82C55A are divided into two groups of 12 signals: group A and group B. One 8-bit control word selects the mode of operation for each group. The group A control bits configure port A (A<7..0>) and the upper 4 bits (nibble) of port C (C<7..4>). The group B control bits configure port B (B<7..0>) and the lower nibble of port C (C<3..0>). These configuration bits are defined in the section *Register Description for the 82C55A* later in this appendix. Because there are four 82C55A PPI devices on the PCI-DIO-96 and PXI-6508, they are referenced as PPI A, PPI B, PPI C, and PPI D when differentiation is required.

On the PCI-DIO-96 and PXI-6508, the three 16-bit counters of the 82C53 are accessed through individual data ports and controlled by one 8-bit control word. The control word selects how the counter data ports are accessed and what mode the counter uses. The configuration bits are defined in the section *Register Description for the 82C53* later in this appendix.

In addition to the 82C55A and 82C53 devices, there are two registers that select which onboard signals are capable of generating interrupts. There are two interrupt signals from each of the 82C55A devices and two interrupt signals from the 82C53 device. Individual enable bits select which of these 10 signals can generate interrupts. Also, a master enable signal determines whether the board can actually send a request to the computer. The configuration bits for these registers are defined in the section *Register Description for the Interrupt Control Registers* later in this appendix.

## Register Map

Table B-1 lists the address map for your DIO board. The PCI-DIO-96 and PXI-6508 use all of the registers. The PCI-6503 uses a subset of the registers, as indicated in the table.

**Table B-1.** Register Address Map

| Register Name | Offset Address (Hex) | Size | Type | Present on the PCI-6503 |
|---|---|---|---|---|
| 82C55A Register Group | | | | |
| PPI A | | | | |
| PORTA Register | 00 | 8-bit | Read-and-write | Yes |
| PORTB Register | 01 | 8-bit | Read-and-write | Yes |
| PORTC Register | 02 | 8-bit | Read-and-write | Yes |
| Configuration Register | 03 | 8-bit | Write-only | Yes |
| PPI B | | | | |
| PORTA Register | 04 | 8-bit | Read-and-write | No |
| PORTB Register | 05 | 8-bit | Read-and-write | No |
| PORTC Register | 06 | 8-bit | Read-and-write | No |
| Configuration Register | 07 | 8-bit | Write-only | No |
| PPI C | | | | |
| PORTA Register | 08 | 8-bit | Read-and-write | No |
| PORTB Register | 09 | 8-bit | Read-and-write | No |
| PORTC Register | 0A | 8-bit | Read-and-write | No |
| Configuration Register | 0B | 8-bit | Write-only | No |
| PPI D | | | | |
| PORTA Register | 0C | 8-bit | Read-and-write | No |
| PORTB Register | 0D | 8-bit | Read-and-write | No |
| PORTC Register | 0E | 8-bit | Read-and-write | No |
| Configuration Register | 0F | 8-bit | Write-only | No |
| 82C53 Register Group | | | | |
| Counter 0 | 10 | 8-bit | Read-and-write | No |
| Counter 1 | 11 | 8-bit | Read-and-write | No |
| Configuration Register | 13 | 8-bit | Write-only | No |
| Interrupt Control Register Group | | | | |
| Register 1 | 14 | 8-bit | Write-only | Yes |
| Register 2 | 15 | 8-bit | Write-only | Yes |
| Interrupt Clear Register | 16 | 8-bit | Write-only | No |

# Register Descriptions

The register descriptions for the devices used on your DIO board are given on the pages that follow. The register description bits labeled with an *X* indicate reserved bits. Always write a 0 to these bits.

## Register Description Format

The remainder of this section discusses each of the DIO board registers in the order shown in Table B-1. Each register group is introduced, followed by a detailed bit description of each register. Individual register descriptions give the address (in hexadecimal), type, data size, and bit map of the register, followed by a description of each bit.

The register bit map shows a diagram of the register with the MSB (bit 7) shown on the left, and the LSB (bit 0) shown on the right. A rectangle with the bit name inside represents each bit.

The bit map for the Interrupt Clear Register states *not applicable, no bits used*. The data is ignored when you write to this register; therefore, any bit pattern will suffice.

## Register Description for the 82C55A

Figure B-1 shows the two control word formats used to completely program the 82C55A. The control word flag (bit 7) determines which control word format is being programmed. When the control word flag is 1, bits 6 through 0 select the I/O characteristics of the 82C55A ports. These bits also select the mode in which the ports are operating; that is, mode 0, mode 1, or mode 2. When the control word flag is 0, bits 3 through 0 select the bit set/reset format of port C.

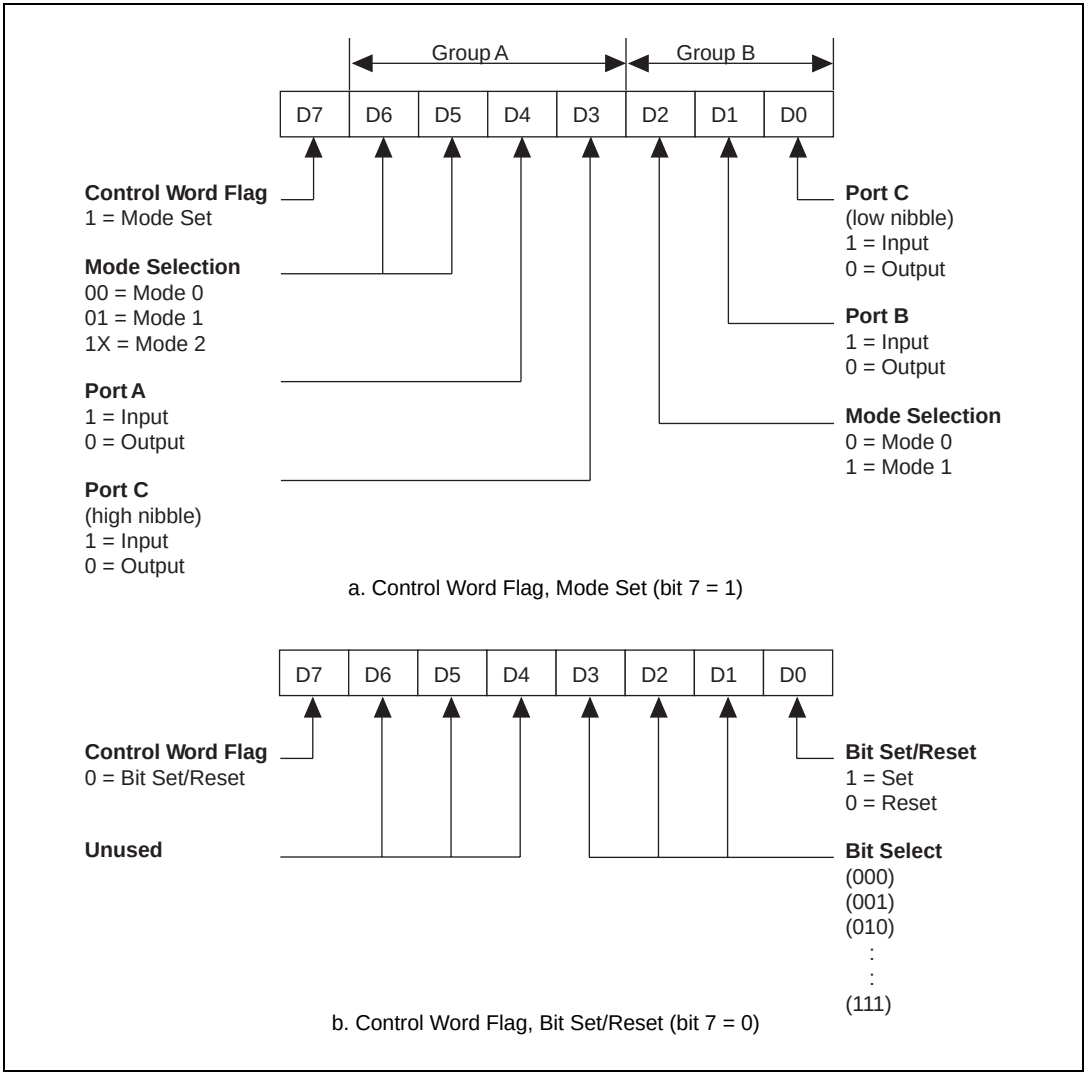



**Figure B-1.** Control Word Formats for the 82C55A

Table B-2 shows the control words for setting or resetting each bit in port C. Notice that bit 7 of the control word is cleared when programming the set/reset option for the bits of port C.

**Table B-2.** Port C Set/Reset Control Words

| Bit Number | Bit Set Control Word | Bit Reset Control Word | Bit Set or Reset in Port C |
|---|---|---|---|
| 0 | 0xxx0001 | 0xxx0000 | xxxxxxxb |
| 1 | 0xxx0011 | 0xxx0010 | xxxxxxbx |
| 2 | 0xxx0101 | 0xxx0100 | xxxxxbxx |
| 3 | 0xxx0111 | 0xxx0110 | xxxxbxxx |
| 4 | 0xxx1001 | 0xxx1000 | xxxbxxxx |
| 5 | 0xxx1011 | 0xxx1010 | xxbxxxxx |
| 6 | 0xxx1101 | 0xxx1100 | xbxxxxxx |
| 7 | 0xxx1111 | 0xxx1110 | bxxxxxxx |

## Register Description for the 82C53 (PCI-DIO-96, PXI-6508 Only)

Figure B-2 shows the control word format used to program the 82C53. Bits 7 and 6 of the control word select the counter to be programmed. Bits 5 and 4 select the mode by which the count data is written to and read from the selected counter. Bits 3, 2, and 1 select the mode for the selected counter. Bit 0 selects whether the counter counts in binary or BCD format. After writing to the Configuration Register to configure a counter, you can read or write the counter itself eight bits at a time, as controlled by the access mode.

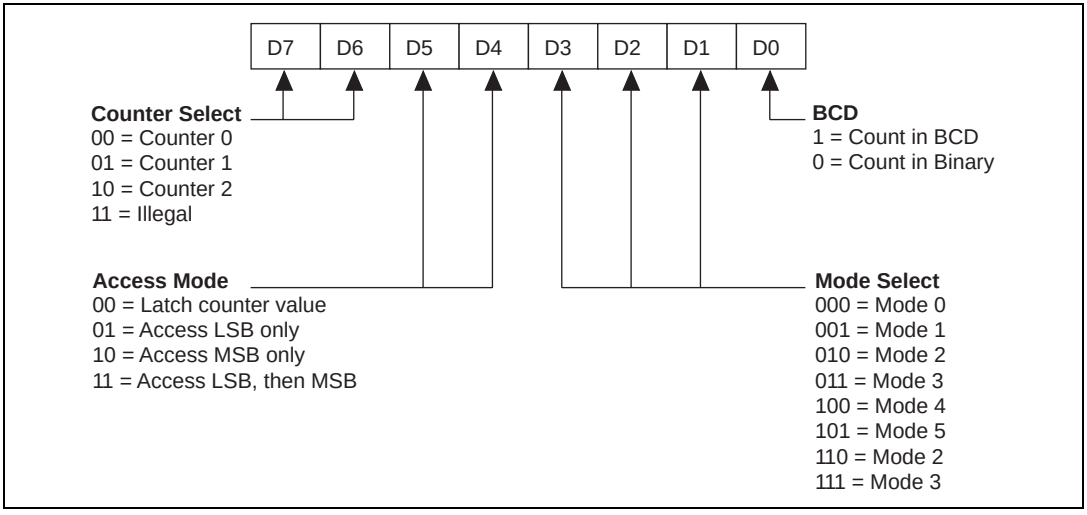



**Figure B-2.** Control Word Format for the 82C53

## Register Description for the Interrupt Control Registers

There are two interrupt control registers on your DIO board. One of these registers has individual enable bits for the two interrupt lines from each of the 82C55A devices. The other register has a master interrupt enable bit and two bits for the timed interrupt circuitry. Of the latter two bits, one bit enables counter interrupts, while the other selects counter 0 or counter 1. The bit maps and signal definitions are listed in this appendix.

## Interrupt Control Register 1

**Address:** Base address + 14 (hex)

**Type:** Write-only

**Word Size:** 8-bit

**Bit Map (PCI-DIO-96/PXI-6508):**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| DIRQ1 | DIRQ0 | CIRQ1 | CIRQ0 | BIRQ1 | BIRQ0 | AIRQ1 | AIRQ0 |

**Bit Map (PCI-6503):**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| X | X | X | X | X | X | AIRQ1 | AIRQ0 |

| Bit | Name | Description |
|---|---|---|
| 7–2 | X | Reserved on the PCI-6503. |
| 7 | DIRQ1 | PPI D Port B Interrupt Enable Bit—If this bit and the INTEN bit in Interrupt Control Register 2 are both set, PPI D sends an interrupt, INTRB, to the computer. If this bit is cleared, PPI D does not send the interrupt INTRB to the computer, regardless of the setting of INTEN. |
| 6 | DIRQ0 | PPI D Port A Interrupt Enable Bit—If this bit and the INTEN bit in Interrupt Control Register 2 are both set, PPI D sends an interrupt, INTRA, to the computer. If this bit is cleared, PPI D does not send the interrupt INTRA to the computer, regardless of the setting of INTEN. |
| 5 | CIRQ1 | PPI C Port B Interrupt Enable Bit—If this bit and the INTEN bit in Interrupt Control Register 2 are both set, PPI C sends an interrupt, INTRB, to the computer. If this bit is cleared, PPI C does not send the interrupt INTRB to the computer, regardless of the setting of INTEN. |
| 4 | CIRQ0 | PPI C Port A Interrupt Enable Bit—If this bit and the INTEN bit in Interrupt Control Register 2 are both set, PPI C sends an interrupt, INTRA, to the computer. If this bit is cleared, PPI C does not send the interrupt INTRA to the computer, regardless of the setting of INTEN. |

| | | |
|---|---|---|
| 3 | BIRQ1 | PPI B Port B Interrupt Enable Bit—If this bit and the INTEN bit in Interrupt Control Register 2 are both set, PPI B sends an interrupt, INTRB, to the computer. If this bit is cleared, PPI B does not send the interrupt INTRB to the computer, regardless of the setting of INTEN. |
| 2 | BIRQ0 | PPI B Port A Interrupt Enable Bit—If this bit and the INTEN bit in Interrupt Control Register 2 are both set, PPI B sends an interrupt, INTRA, to the computer. If this bit is cleared, PPI B does not send the interrupt INTRA to the computer, regardless of the setting of INTEN. |
| 1 | AIRQ1 | PPI A Port B Interrupt Enable Bit—If this bit and the INTEN bit in Interrupt Control Register 2 are both set, PPI A sends an interrupt, INTRB, to the computer. If this bit is cleared, PPI A does not send the interrupt INTRB to the computer, regardless of the setting of INTEN. |
| 0 | AIRQ0 | PPI A Port A Interrupt Enable Bit—If this bit and the INTEN bit in Interrupt Control Register 2 are both set, PPI A sends an interrupt, INTRA, to the computer. If this bit is cleared, PPI A does not send the interrupt INTRA to the computer, regardless of the setting of INTEN. |

## Interrupt Control Register 2

**Address:** Base address + 15 (hex)

**Type:** Write-only

**Word Size:** 8-bit

**Bit Map (PCI-DIO-96/PXI-6508):**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| X | X | X | X | X | INTEN | CTRIRQ | CTR1 |

**Bit Map (PCI-6503):**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| X | X | X | X | X | INTEN | X | X |

| Bit | Name | Description |
|---|---|---|
| 7–3 | X | Reserved. |
| 2 | INTEN | Interrupt Enable Bit—If this bit is set, the DIO board can interrupt the computer. If this bit is cleared, the DIO board cannot generate interrupts to the computer, regardless of the status of the bits in Interrupt Control Register 2. |
| 1–0 | X | Reserved on the PCI-6503. |
| 1 | CTRIRQ | Counter Interrupt Enable Bit—If this bit is set, the 82C53 counter outputs can interrupt the computer. If this bit is cleared, the counter outputs have no effect. To avoid a spurious interrupt, keep INTEN low when you set CTRIRQ; that is, set CTRIRQ before setting INTEN. |
| 0 | CTR1 | Counter Select Bit—If this bit is set, the output from counter 1 of the 82C53 is connected to the interrupt request circuitry. In this mode, counter 0 of the 82C53 acts as a frequency scaler for counter 1, which generates the interrupt. If CTR1 is cleared, the output from counter 0 of the 82C53 is connected to the interrupt request circuitry. In this mode, counter 0 generates the interrupt. For more information, see the *Interrupt Programming Example* section for the 82C53 in this appendix. |

## Interrupt Clear Register (PCI-DIO-96, PXI-6508 Only)

The interrupt clear register has no bits associated with it. Use this register to reset the state of the interrupt request signal once the interrupt routine has been entered. To clear the interrupt, perform an 8-bit write to this register address; the data is irrelevant.

**Address:** Base address + 16 (hex)

**Type:** Write-only

**Word Size:** 8-bit

**Bit Map:**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| X | X | X | X | X | X | X | X |

| Bit | Name | Description |
|---|---|---|
| 7–0 | X | Don't care bit. |

# Programming

This section contains instructions on how to operate your DIO board circuitry, and examples of the programming steps necessary to execute an operation. If you are not using NI-DAQ, you must first initialize your board. The initialization steps are different for PC and Macintosh users, so refer to the section pertaining to your platform. The PCI-DIO-96 and PCI-6503 supports the PC and Macintosh; the PXI-6508 supports only the PC (that is, the PXI or CompactPCI controller).

Programming your DIO board involves writing to and reading from registers on the board. You will find a listing of these registers in the *Register Map and Description* section of this appendix.

## PCI Local Bus

The PCI-DIO-96, PXI-6508, and PCI-6503 are fully compatible with the *PCI Local Bus Specification, Revision 2.1*, from the PCI Special Interest Group (SIG). The PXI-6508 is fully compliant with the National Instruments *PXI Specification, Revision 1.0*. All three boards use the PCI Local Bus to move data. The PCI Local Bus is a high performance, 32-bit bus with multiplexed address and data lines. The PCI system arbitrates and assigns resources through software, freeing you from manually setting switches and jumpers. Bus-related resources must be configured before you attempt to execute a register-level program. This entails assigning a base address and interrupt channel to your DIO board.

You can use PCI local bus boards on both PC-compatible and Macintosh computers. However, due to the differences in those systems, configuration will be different.

## Programming Examples

The programming examples in this section demonstrate the programming steps needed to perform several different operations. The instructions are language independent; that is, they tell you to read or write a given register or to detect if a given bit is set or cleared, without presenting the actual code. The information given is not intended to be used without proper modification in a practical solution.

Before you can implement any of the examples into a real application, you must know the base memory address for your board. To generate and process any interrupts, you must write and install an applicable interrupt service routine.

☞ **Note** ***In this appendix all numbers preceded by 0x are hexadecimal.***

Common terms that you will see used in the programming examples are listed below:

| | |
|---|---|
| *Port A* | Address of PPI A Port A Register (Base Address + 0x00) |
| *Port B* | Address of PPI A Port B Register (Base Address + 0x01) |
| *Port C* | Address of PPI A Port C Register (Base Address + 0x02) |
| *8255Cnfg* | Address of PPI A Configuration Register (Base Address + 0x03) |
| *Ctr0* | Address of 82C53 Counter 0 Register (Base Address + 0x10) |
| *Ctr1* | Address of 82C53 Counter 1 Register (Base Address + 0x11) |
| *CntrCnfg* | Address of 82C53 Configuration Register (Base Address + 0x13) |
| *IREG1* | Address of Interrupt Control Register 1 (Base Address + 0x14) |
| *IREG2* | Address of Interrupt Control Register 2 (Base Address + 0x15) |
| *Write (address, data)* | Generic function call for a memory space Write of data to address |
| *Read (address)* | Generic function call for a memory space Read from address |
| *CWrite (offset, data)* | PCI configuration space write of data to PCI configuration space offset |

## PCI Initialization for the PC

To program at the register level without NI-DAQ, you must know the PCI-DIO-96 or PXI-6508 base memory address and, if using interrupts, install an interrupt handler. Writing an interrupt handler is solely left to you and is not discussed in this manual. In order for the board to operate properly, the PCI MITE ASIC must be configured. Ordinarily, NI-DAQ performs this function, but if you are not using NI-DAQ, then you must configure the PCI MITE ASIC.

The following sections explain how to accomplish this. The references made to PCI BIOS[1] calls are left to you to implement.

In order to configure the PCI MITE chip you must first write an algorithm that finds and stores all configuration information about the board. You can do this by using PCI BIOS calls to search PCI configuration space for the National Instruments vendor ID (0x1093) and PCI-DIO-96 device ID (0x0160), PXI-6508 device ID (0x13c0), or PCI-6503 device ID (0x17d0). If a board is found, the algorithm can store all the board's configuration information into a data structure. Base Address Register 0 (BAR0) corresponds to the base address of the PCI MITE, while Base Address Register 1 (BAR1) is the base address of the board registers. The size of each of these windows is 4 KB. Both addresses will most likely be mapped above 1 MB in the memory map. This means that in order to communicate with the board you must know how to perform memory cycles to extended memory. Information is provided to re-map the board under 1 MB in the memory map, which makes communicating with the board simpler. PCI BIOS read and write calls are used to accomplish this. Use the pseudocode in this section to re-map the board below 1 MB. If you choose not to re-map the board, you must still perform Steps 4 and 5. All values in this example are 32 bits:

1. Write the address to which you want to re-map the PCI MITE to *PCI configuration space offset 0x10 (BAR0)*.
2. Write the value *0x0000aeae* to offset *0x340* from the new PCI MITE address.
3. Write the address to which you want to re-map the board (other than the PCI MITE) to *PCI configuration space offset 0x14 (BAR1)*.
4. Create the window data value by masking the new board address:

   window data value = ((0xffffff00 *and* new board address) *or* (0x00000080))

   If you are not remapping the board, then the new board address is the value in *BAR1*.
5. Write the window data value to offset 0xc0 from the new PCI MITE address.
   If you are not remapping the board, then the new PCI MITE address is the value in *BAR0*.

The following pseudocode re-maps the PCI MITE to memory address *0xd0000* and the board to memory address *0xd1000*.

```
CWrite(0x10,0x000d0000)
Write(0xd0340,0x0000aeae)
CWrite(0x14,0x000d1000)
Write(0xd00c0,0x000d1080)
```

The new base address for the PCI-DIO-96 or PXI-6508 would now be *0xd1000*, for this example. It is important that the memory range to which you re-map the board is not being used by another device or system resource. You can exclude this memory from use with a memory manager.

[1] You can obtain more information on PCI BIOS calls from the PCI SIG on the World Wide Web (`http://www.pcisig.com`).

# PCI Initialization for the Macintosh (PCI-DIO-96, PCI-6503 Only)

## Programming Options

To program at the register level, you must know the PCI-DIO-96 and PCI-6503 base memory address and you must install an interrupt handler if you want to generate interrupts. Both of these operations are difficult tasks. To make this process easier, National Instruments provides a driver toolkit and additional NI-DAQ functions to perform these operations.

You have three options to program the PCI-DIO-96. The following sections describe these options. Only the third option applies to the PCI-6503.

### Using NI-DAQ and the Driver Toolkit (PCI-DIO-96 and NI-DAQ 4.90 Only)

Included on the NI-DAQ installation media is a toolkit for creating plug-in drivers for most of the devices which NI-DAQ controls. Using this toolkit, you can write a plug-in driver for your PCI board, but continue to use NI-DAQ for any other boards that are installed in your system. When you develop a driver using the toolkit, your driver plug-in has access to all the information and support functions it needs to control the device and respond to interrupts. When you use the toolkit, your application is divided into two parts—a driver and an interface to the driver. You use the driver to control the hardware and the interface to control the driver. You can install the driver toolkit by launching the NI-DAQ installer, choosing the alternate installations option (see the installer for help), and dragging the toolkit icon to your disk. Documentation for the toolkit is included in the toolkit.

### Performing Simple Accesses (PCI-DIO-96 Only)

To perform simple input and output using your PCI board without using the drivers included in NI-DAQ or writing your own drivers, you can use the `Get_DAQ_Device_Info` call to do simple accesses with the board. If you want to use interrupts, you must work directly with the Macintosh operating system (Mac OS), and you could inadvertently corrupt portions of NI-DAQ. Therefore, National Instruments recommends this option only if you are not generating interrupts.

### Developing Your Own Configuration Program

National Instruments does not support developing your own interrupt method. To do this, consult the following documents:

- *Designing PCI Cards and Drivers for Power Macintosh Computers*
- *Inside Macintosh: Devices*
- *Inside Macintosh: Memory*
- *Inside Macintosh: Operating System Utilities*
- *Inside Macintosh: Processes*
- *Inside Macintosh: Power PC System Software*

When developing your own configuration program, because NI-DAQ has not configured your board, you will need to perform the following code sequence to activate the board. Using the documents listed above, you must retrieve the `deviceNode` parameter from the Name Registry.

```
#include <pci.h>

void  *configureCard(RegEntryIDPtr deviceNode);
void  *configureCard
      (  RegEntryIDPtr     deviceNode
      )
      {  unsigned short    pciCommandRegister;
         unsigned long     cardBaseAddress,
                           miteBaseAddress;


         //         configure the i/o space of the board such
         // that it is memory mapped.
         ExpMgrConfigReadWord(deviceNode,
                    ((LogicalAddress) 0x00000004L),
         &pciCommandRegister);
         ExpMgrConfigWriteWord(deviceNode,
                    ((LogicalAddress) 0x00000004L),
                    (pciCommandRegister | 0x0002));

         //         get the base addresses for the board.
         ExpMgrConfigReadLong(deviceNode,
                    ((LogicalAddress) 0x00000010L),
                    &miteBaseAddress);
         ExpMgrConfigReadLong(deviceNode,
                    ((LogicalAddress) 0x00000014L),
                    &cardBaseAddress);
```

```
        //          activate the standard i/o window.
        *((unsigned long *) (miteBaseAddress +
                    0x000000c0L)) =
                    EndianSwap32Bit(((cardBaseAddress &
                    0xffffff00L) | 0x00000080L));

        //          return the base address of the board.
        return ((void *) cardBaseAddress);
}
```

# Programming Considerations for the 82C55A

## Modes of Operation

The three basic modes of operation for the 82C55A are as follows. Ports A and B can operate in different modes:

- Mode 0—Basic I/O—This mode can be used for simple input and output operations for each port. No handshaking is required; a specified port simply writes to or reads from data. Mode 0 has the following features:
  - Two 8-bit ports (A and B) and two 4-bit ports (upper and lower nibbles of port C).
  - Any port can be input or output.
  - Outputs are latched, but inputs are not latched.
- Mode 1—Strobed I/O—This mode transfers data that is synchronized by handshaking signals. Ports A and B use the eight lines of port C to generate or receive the handshake signals. This mode divides the ports into two groups (group A and group B) and includes the following features:
  - Each group contains one 8-bit data port (port A or port B) and one 3-bit control/data port (upper or lower portion of port C).
  - The 8-bit data ports can be either input or output; both are latched.
  - The 3-bit ports are used for control and status of the 8-bit data ports.
  - Interrupt generation and enable/disable functions are available.
- Mode 2—Bidirectional bus—This mode can be used for communication over a bidirectional 8-bit bus. Handshaking signals are used in a manner similar to mode 1. Mode 2 is available for use in group A only (port A and the upper portion of port C). Other features of this mode include the following:
  - One 8-bit bidirectional port (port A) and a 5-bit control/status port (port C).
  - Latched inputs and outputs.
  - Interrupt generation and enable/disable functions.

The 82C55A also has a single bit set/reset feature for port C, which is programmed by the 8-bit control word. Any of the eight bits of port C can be set or reset with one control word. This feature generates control signals for port A and port B when these ports are operating in mode 1 or mode 2.

## Mode 0–Basic I/O

You can use mode 0 for simple I/O functions (no handshaking) for each of the three ports and assign each port as an input or an output port. Table B-3 shows the 16 possible I/O configurations. Notice that bit 7 of the control word is set when programming the mode of operation for each port.

**Table B-3.** Mode 0 I/O Configurations

| Number | Control Word Bit 76543210 | Group A | | Group B | |
|---|---|---|---|---|---|
| | | Port A | Port C[1] | Port B | Port C[2] |
| 0 | 10000000 | Output | Output | Output | Output |
| 1 | 10000001 | Output | Output | Output | Input |
| 2 | 10000010 | Output | Output | Input | Output |
| 3 | 10000011 | Output | Output | Input | Input |
| 4 | 10001000 | Output | Input | Output | Output |
| 5 | 10001001 | Output | Input | Output | Input |
| 6 | 10001010 | Output | Input | Input | Output |
| 7 | 10001011 | Output | Input | Input | Input |
| 8 | 10010000 | Input | Output | Output | Output |
| 9 | 10010001 | Input | Output | Output | Input |
| 10 | 10010010 | Input | Output | Input | Output |
| 11 | 10010011 | Input | Output | Input | Input |
| 12 | 10011000 | Input | Input | Output | Output |
| 13 | 10011001 | Input | Input | Output | Input |
| 14 | 10011010 | Input | Input | Input | Output |
| 15 | 10011011 | Input | Input | Input | Input |

[1] Upper nibble of port C
[2] Lower nibble of port C

### Mode 0 Basic I/O Programming Example

The following example shows how to configure PPI A for mode 0 input and output.

```
Write (8255Cnfg,0x80)    Set mode 0—ports A, B, and C are outputs
Write (PortA, Data)      Write data to port A
Write (PortB, Data)      Write data to port B
Write (PortC, Data)      Write data to port C

Write (8255Cnfg,0x90)    Set mode 0—port A is Input;ports B and
                         C are outputs
Write (PortB, Data)      Write data to port B
Read (PortA)             Read data from port A
```

## Mode 1–Strobed Input

☞ **Note** ***For mode 1 examples, you must configure the don't care bits appropriately in the control word if you want to use the other ports in combination with the example.***

In mode 1, the digital I/O bits are divided into two groups: group A and group B. Each of these groups contains one 8-bit port and one 3-bit control/data port. The 8-bit port can be either an input or an output port, and the 3-bit port is used for control and status information for the 8-bit port. The transfer of data is synchronized by handshaking signals in the 3-bit port.

The control word written to the Configuration Register to configure port A for input in mode 1 is shown in Figure B-3. You can use bits PC6 and PC7 of port C as extra input or output lines.

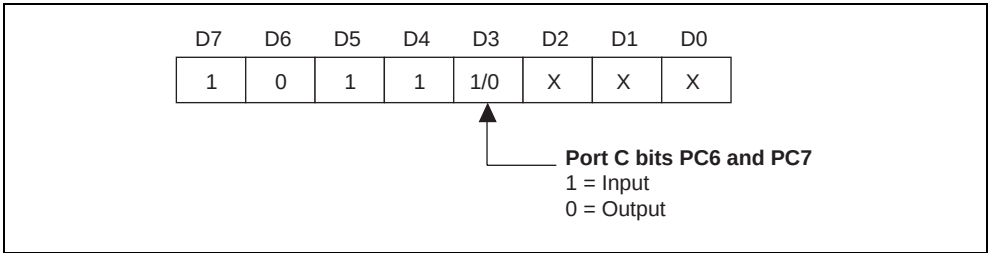



**Figure B-3.** Control Word to Configure Port A for Mode 1 Input

Figure B-4 shows the control word written to the Configuration Register to configure port B for input in mode 1. Notice that port B does not have extra input or output lines left from port C when ports A and B are both enabled for handshaking.

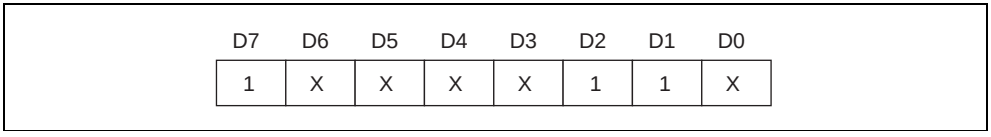



**Figure B-4.** Control Word to Configure Port B for Mode 1 Input

During a mode 1 data read transfer, read port C to obtain the status of the handshaking lines and interrupt signals. See the *Port C Status-Word Bit Definitions for Input (Ports A and B)*, *Port C Status-Word Bit Definitions for Output (Ports A and B)*, and *Port C Status-Word Bit Definitions for Bidirectional Data Path (Port A Only)* sections in this appendix for detailed definitions.

## Port C Status-Word Bit Definitions for Input (Ports A and B)

**Address:** Base address + 02 (hex) for PPI A
Base address + 06 (hex) for PPI B
Base address + 0A (hex) for PPI C
Base address + 0E (hex) for PPI D

**Type:** Read and write

**Word Size:** 8-bit

**Bit Map:**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| I/O | I/O | IBFA | INTEA | INTRA | INTEB | IBFB | INTRB |

| Bit | Name | Description |
|---|---|---|
| 7–6 | I/O | Input/Output—These bits can be used for general-purpose I/O when port A is in mode 1 input. If these bits are configured for output, the port C bit set/reset function must be used to manipulate them. |
| 5 | IBFA | Input Buffer Full for Port A—A high setting indicates that data has been loaded into the input latch for port A. |
| 4 | INTEA | Interrupt Enable Bit for Port A—Setting this bit enables the INTRA flag from port A of the 82C55A. Control INTEA by setting/resetting PC4. |
| 3 | INTRA | Interrupt Request Status for Port A—This status flag, which operates only when INTEA is high, indicates that port A has acquired data and is ready to be read. If you have enabled interrupts (by setting INTEN and the appropriate bit in Interrupt Control Register 2), this status flag also indicates that an interrupt request is pending for port A. |
| 2 | INTEB | Interrupt Enable Bit for Port B—Setting this bit enables the INTRB flag from port B of the 82C55A. Control INTEB by setting/resetting PC2. |
| 1 | IBFB | Input Buffer Full for Port B—A high setting indicates that data has been loaded into the input latch for port B. |

| | | |
|---|---|---|
| 0 | INTRB | Interrupt Request Status for Port B—Interrupt Request Status for Port B. This status flag, which operates only when INTEA is high, indicates that port B has acquired data and is ready to be read. If you have enabled interrupts (by setting INTEN and the appropriate bit in Interrupt Control Register 2), this status flag also indicates that an interrupt request is pending for port B. |

At the digital I/O connector, port C has the pin assignments shown in Figure B-5 when in mode 1 input. Notice that the status of STBA* and the status of STBB* are not included in the port C status word.

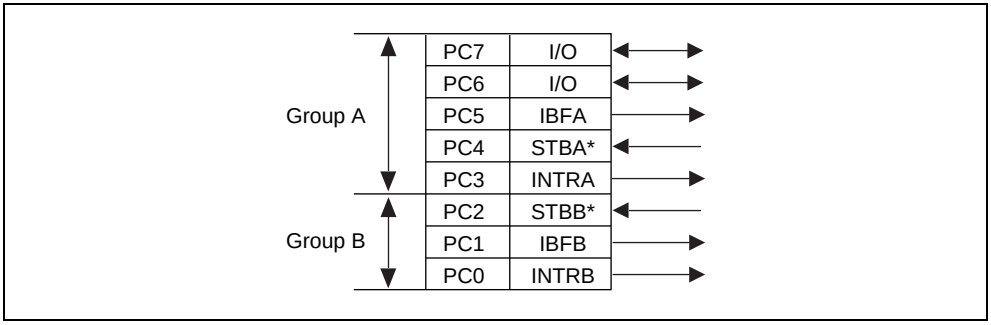



**Figure B-5.** Port C Pin Assignments on I/O Connector when Port C Configured for Mode 1 Input

### Mode 1 Strobed Input Programming Example

The following example shows how to configure PPI A for mode 1 input.

```
Write (8255Cnfg, 0xB0) Set mode 1—port A is an input
Write (8255Cnfg, 0x09) Set PC4 to enable the INTRA status flag
Loop until the INTRA (PC3) and IBFA (PC5) status flags are set,
                       indicating that the 82C55A is ready for a
                       transfer and that the input buffer is full
Read (PortA)           Read data from port A
```

## Mode 1–Strobed Output

☞ **Note** ***For mode 1 examples, you must configure the don't care bits appropriately in the control word if you want to use the other ports in combination with the example.***

The control word written to the Configuration Register to configure port A for output in mode 1 is shown in Figure B-6. You can use bits PC4 and PC5 of port C as extra input or output lines.

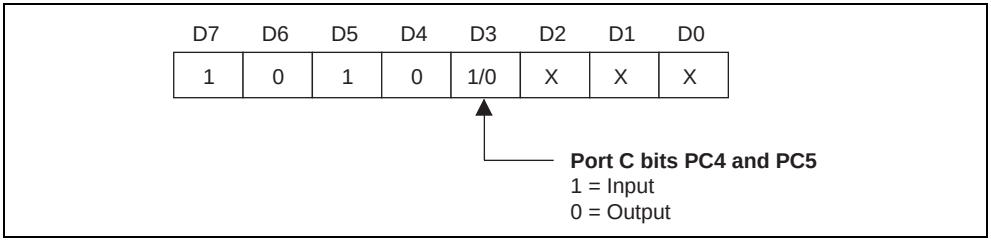



**Figure B-6.** Control Word to Configure Port A for Mode 1 Output

The control word written to the Configuration Register to configure port B for output in mode 1 is shown in Figure B-7. Notice that port B does not have extra input or output lines left from port C when ports A and B are both configured for handshaking.

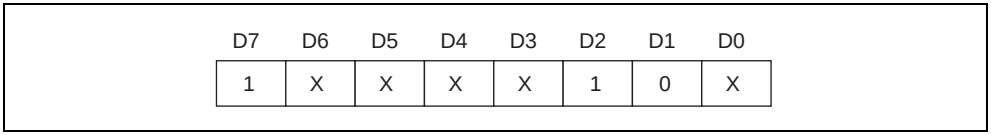



**Figure B-7.** Control Word to Configure Port B for Mode 1 Output

During a mode 1 data write transfer, you can obtain the status of the handshaking lines and interrupt signals by reading port C. Notice that the bit definitions are different for a write and a read transfer.

## Port C Status-Word Bit Definitions for Output (Ports A and B)

**Address:** Base address + 02 (hex) for PPI A
Base address + 06 (hex) for PPI B
Base address + 0A (hex) for PPI C
Base address + 0E (hex) for PPI D

**Type:** Read and write

**Word Size:** 8-bit

**Bit Map:**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| OBFA* | INTEA | I/O | I/O | INTRA | INTEB | OBFB* | INTRB |

| Bit | Name | Description |
|---|---|---|
| 7 | OBFA* | Output Buffer Full for Port A—A low setting indicates that the CPU has written data to port A. |
| 6 | INTEA | Interrupt Enable Bit for Port A—Setting this bit enables the INTRA flag from port A of the 82C55A. Control this bit by setting/resetting PC6. |
| 5–4 | I/O | Input/Output—These bits can be used for general-purpose I/O when port A is in mode 1 output. If these bits are configured for output, you must use the port C bit set/reset function to manipulate them. |
| 3 | INTRA | Interrupt Request Status for Port A—This status flag, which operates only when INTEA is high, indicates that port A has acquired data and is ready to be read. If you have enabled interrupts (by setting INTEN and the appropriate bit in Interrupt Control Register 2), this status flag also indicates that an interrupt request is pending for port A. |
| 2 | INTEB | Interrupt Enable Bit for Port B—Setting this bit enables the INTRB flag from port B of the 82C55A. Control this bit by setting/resetting PC2. |
| 1 | OBFB* | Output Buffer Full for Port B—A low setting indicates that the CPU has written data to port B. |

| | | |
|---|---|---|
| 0 | INTRB | Interrupt Request Status for Port B—This status flag, which operates only when INTEA is high, indicates that port B has acquired data and is ready to be read. If you have enabled interrupts (by setting INTEN and the appropriate bit in Interrupt Control Register 2), this status flag also indicates that an interrupt request is pending for port B. |

At the digital I/O connector, port C has the pin assignments shown in Figure B-8 when in mode 1 output. Notice that the status of ACKA* and ACKB* are not included when port C is read.

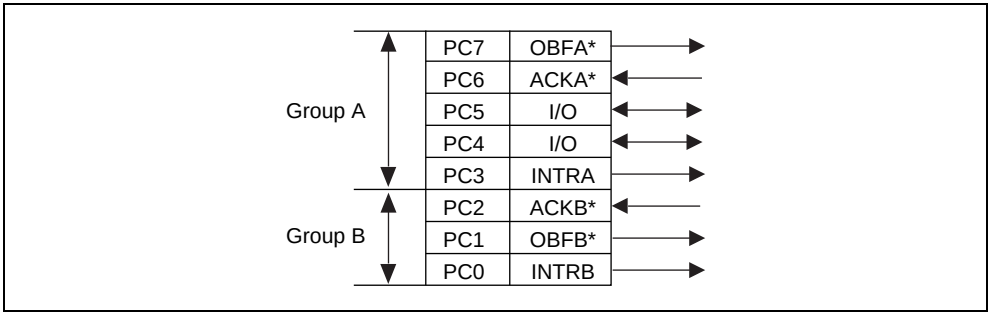



**Figure B-8.** Port C Pin Assignments on I/O Connector when Port C Configured for Mode 1 Output

### Mode 1 Strobed Output Programming Example

The following example shows how to configure PPI A for mode 1 output.

```
Write (8255Cnfg, 0xA0)   Set mode 1—port A is an output
Write (8255Cnfg, 0x0D)   Set PC6 to enable the INTRA status flag
Loop until the INTRA (PC3) and OBFA* (PC7) status flags are set,
                         indicating that the 82C55A is ready for a
                         transfer and that the output buffer is not
                         full
Write (PortA, Data)      Write data to port A
```

## Mode 2–Bidirectional Bus

☞ **Note** ***For mode 2 examples, you must configure the don't care bits appropriately in the control word if you want to use the other ports in combination with the example.***

Mode 2 has an 8-bit bus that can transfer both input and output data without changing the configuration. The data transfers are synchronized with handshaking lines in port C. This mode uses only port A; however, port B can be used in either mode 0 or mode 1 while port A is configured for mode 2.

The control word written to the Configuration Register to configure port A as a bidirectional data bus in mode 2 is shown in Figure B-9. If port B is configured for mode 0, you can use PC2, PC1, and PC0 of port C as extra input or output lines.

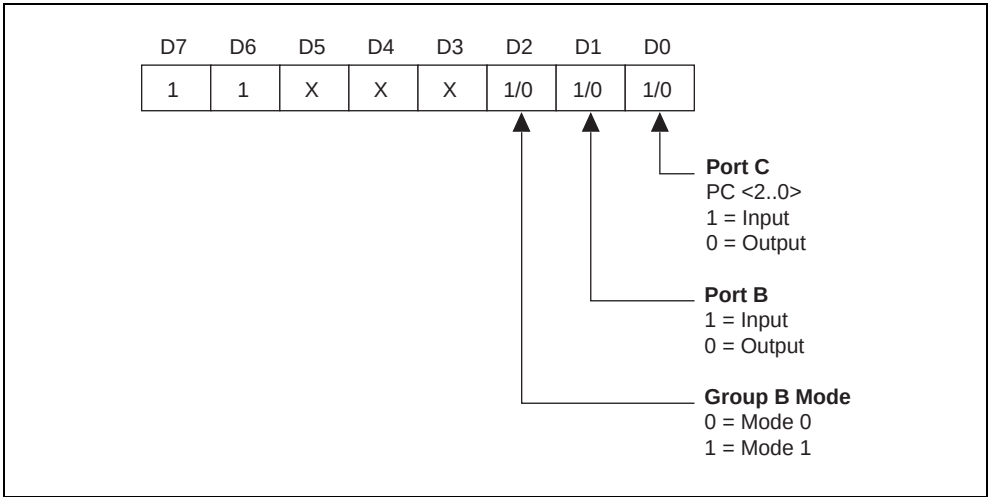



**Figure B-9.** Control Word to Configure Port A as Mode 2 Bidirectional Data Bus

During a mode 2 data transfer, you can obtain the status of the handshaking lines and interrupt signals by reading port C. The port C status-word bit definitions for a mode 2 transfer are shown as follows.

# Port C Status-Word Bit Definitions for Bidirectional Data Path (Port A Only)

**Address:** Base address + 02 (hex) for PPI A
Base address + 06 (hex) for PPI B
Base address + 0A (hex) for PPI C
Base address + 0E (hex) for PPI D

**Type:** Read and write

**Word Size:** 8-bit

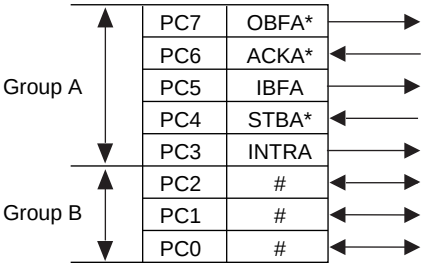
**Bit Map:**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| OBFA* | INTE1 | IBFA | INTE2 | INTRA | I/O | I/O | I/O |

| Bit | Name | Description |
|---|---|---|
| 7 | OBFA* | Output Buffer Full for Port A—A low setting indicates that the CPU has written data to port A. |
| 6 | INTE1 | Interrupt Enable Bit for Port A Output Interrupts—Setting this bit enables the INTRA flag from port A of the 82C55A for output. Control this bit by setting/resetting PC6. |
| 5 | IBFA | Input Buffer Full for Port A—A high setting indicates that data has been loaded into the input latch of port A. |
| 4 | INTE2 | Interrupt Enable Bit for Port A Input Interrupts—Setting this bit enables the INTRA flag from port A of the 82C55A for input. Control this bit by setting/resetting PC4. |
| 3 | INTRA | Interrupt Request Status for Port A—This status flag, which operates only when INTE1 or INTE2 is high, indicates that port A is ready to be read or written; check the IBF and OBFA* flags to determine which. If you have enabled interrupts (by setting INTEN and the appropriate bit in Interrupt Control Register 2), the INTRA status flag also indicates that an interrupt request is pending for port A. |

| | | |
|---|---|---|
| 2-0 | I/O | Input/Output—Use these bits for general-purpose I/O lines if group B is configured for mode 0. If group B is configured for mode 1, refer to the bit explanations shown in the preceding mode 1 sections. |

Figure B-10 shows the port C pin assignments on the digital I/O connector when port C is configured for mode 2. Notice that the status of STBA* and the status of ACKA* are not included in the port C status word.




# The three port C lines associated with group B function based on the mode selected for group B; that is, if group B is configured for mode 0, PC<2..0> function as general-purpose I/O, but if group B is configured for mode 1 input or output, PC<2..0> function as handshaking lines as shown in the preceding mode 1 sections.

**Figure B-10.** Port C Pin Assignments on I/O Connector when Port C is Configured for Mode 2

## Mode 2 Bidirectional Bus Programming Example

The following example shows how to configure PPI A for mode 2 input and output.

```
Write (8255Cnfg, 0xC0)           Set mode 2—port A is bidirectional
Write (8255Cnfg, 0x09)           Set PC4 to enable the INTRA status
                                 flag for input
Write (8255Cnfg, 0x0D)           Set PC6 to enable the INTRA status
                                 flag for output
Loop until the INTRA (PC3) status flag is set, indicating the 82C55
                                 is ready for a transfer
If IBFA (PC5) is set, read (PortA)If input buffer is full, read data
If OBFA* (PC7) is set,
   write (PortA, data)           If output buffer is not full, write
                                 data
```

## Interrupt Handling

You must set the INTEN bit of Interrupt Control Register 2 to enable interrupts. Clear this bit first to disable unwanted interrupts. After all sources of interrupts have been disabled or placed in an inactive state, you can set INTEN. You must set INTEN before you generate an interrupt for proper operation.

To interrupt the computer using one of the 82C55A devices, program the selected 82C55A for the I/O mode desired. In mode 1, set either the INTEA or the INTEB bit to enable interrupts from port A or port B, respectively. In mode 2, set either INTE1 or INTE2 for interrupts on output or input transfers, respectively. The INTE1 and INTE2 interrupt outputs are cascaded into a single interrupt output for port A. After you enable interrupts from the 82C55A, set the appropriate enable bit for the selected 82C55A; for example, if you select both mode 2 interrupts for PPI C, set CIRQ0 to interrupt the computer.

To interrupt the computer using one of the 82C53 counter outputs on the PCI-DIO-96 or PXI-6508, program the counters as described in the *Interrupt Programming Example* section later in this appendix.

You can use external signals to generate interrupts when port A or port B is in mode 0 and the low nibble of port C is configured for input. If port A is in mode 0, use PC3 to generate an interrupt; if port B is in mode 0, use PC0 to generate an interrupt. After you have configured the selected 82C55A, you must set the corresponding interrupt enable bit in Interrupt Control Register 1. If you are using PC3, set *x*IRQ0; if you are using PC0, set *x*IRQ1, where *x* is the letter corresponding to the PPI you want to generate interrupts (A–D). When the external signal becomes logic high, an interrupt request occurs. To disable the external interrupt, the interrupt service routine that you have written should acknowledge the interrupt. On the PCI-DIO-96 and PXI-6508, the interrupt service routine should also write the interrupt clear register.

## Interrupt Programming Examples for the 82C55A

The following examples show the process required to enable interrupts for several different operating modes. You must write and install an interrupt service routine in order to process the interrupt and gain any useful knowledge from it. You should clear all interrupt sources and interrupt enable bits first to disable unwanted interrupts.

### Mode 1 Strobed Input Programming Example

The following example shows how to set up interrupts for mode 1 input for port A.

```
Write (8255Cnfg, 0xB0)   Set mode 1—port A is an input
Write (8255Cnfg, 0x09)   Set PC4 to enable interrupts from
                         the 82C55A
Write (IREG2, 0x04)      Set INTEN bit
Write (IREG1, 0x01)      Set AIRQ0 to enable PPI A,
                         port A interrupts
```

### Mode 1 Strobed Output Programming Example

The following example shows how to set up interrupts for mode 1 output for port A.

```
Write(8255Cnfg, 0xA0)    Set mode 1—port A is an output
Write(8255Cnfg, 0x0D)    Set PC6 to enable interrupts from 82C55A
Write(IREG2, 0x04)       Set INTEN bit
Write(IREG1, 0x01)       Set AIRQ0 to enable PPI A,
                         port A interrupts
```

### Mode 2 Bidirectional Bus Programming Example

The following example shows how to set up interrupts for mode 2 output transfers.

```
Write (8255Cnfg, 0xC0)   Set mode 2—port A is bidirectional
Write (8255Cnfg, 0x0D)   Set PC6 to enable interrupt from 82C55A
Write (IREG2, 0x04)      Set INTEN bit
Write (IREG1, 0x01)      Set AIRQ0 to enable PPI A,
                         port A interrupts
```

The following example shows how to set up interrupts for mode 2 input transfers.

```
Write (8255Cnfg, 0xC0)   Set mode 2—port A is bidirectional
Write (8255Cnfg, 0x09)   Set PC4 to enable interrupt from 82C55A
Write (IREG2, 0x04)      Set INTEN bit
Write (IREG1, 0x01)      Set AIRQ0 to enable PPI A,
                         port A interrupts
```

## Programming Considerations for the 82C53

The PCI-DIO-96 and PXI-6508 contain an 82C53 programmable interval timer. A general overview of the 82C53 and how it is configured follows.

## General Information

The 82C53 contains three counter/timers, each of which can operate in one of six different modes. However, only counter 0 and counter 1 are configured for operation; counter 2 is not connected, nor is it available on the external I/O connector. In addition, the counter gates are wired in such a way (tied to logic high) that modes 1 and 5 are unusable; the recommended counter mode is mode 2.

The source for counter 0 is a 2 MHz clock. If you use counter 0 to interrupt the computer, configure the counter for rate generation, or mode 2. If you use counter 1 to interrupt the computer, counter 0 is a frequency scale that feeds the source input for counter 1. In this case, configure both counters for rate generation, or mode 2.

To determine the time between pulses generated by counter 0, multiply the load value by 500 ns (1/(2 MHz)). To determine the time between pulses generated by counter 1, multiply the load value by the time between pulses of counter 0. A sample configuration procedure is presented in the next section.

## Interrupt Programming Example

The following example shows how to set up counter 0 to generate interrupts:

```
Write(IREG1, 0x00)      Disable all 82C55A interrupts
Write(IREG2, 0x00)      Disable counter interrupts
Write(CntrCnfg, 0x34)   Set counter 0 to mode 2
Write(IREG2, 0x02)      Enable counter interrupts and select the
                        output from counter 0 before enabling board
                        interrupts
Write(IREG2, 0x06)      Enable board interrupts
Write(Ctr0, Data0)      Send the least significant byte
                        of the counter data to counter 0
Write(Ctr0, Data1)      Send the most significant byte
                        of the counter data to counter 0
```

The counter begins counting as soon as the most significant byte is written. When you are ready to exit your program, disable the counter and interrupts as shown below.

```
Write(Cnfg, 0x30)       Turn off counter 0
Write(IREG2, 0x00)      Disable all PCI-DIO-96/PXI-6508
                        interrupts
```

☞ **Note** ***In order for any of the interrupts to be processed, you must write and install an interrupt service routine. Failure to do so could cause the system to fail upon the interrupt generation.***

# C

# MSM82C55A Data Sheet[1]

This appendix contains a manufacturer data sheet for the MSM82C55A CMOS programmable peripheral interface from OKI Semiconductor. This interface is used on the PCI-DIO-96, PXI-6508, and PCI-6503.

[1] Copyright© OKI Semiconductor. 1995. Reprinted with permission of copyright owner. All rights reserved. OKI Semiconductor. *Microprocessor Data Book 1995*.

**OKI semiconductor**

# MSM82C55A-2RS/GS/VJS

## CMOS PROGRAMMABLE PERIPHERAL INTERFACE

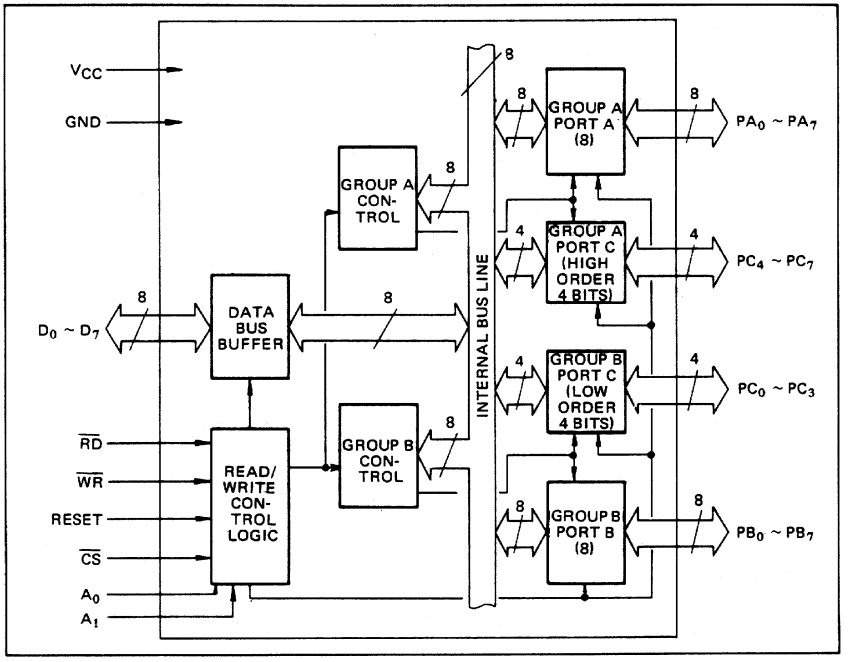
### GENERAL DESCRIPTION

The MSM82C55A is a programmable universal I/O interface device which operates as high speed and on low power consumption due to 3 $\mu$ silicon gate CMOS technology. It is the best fit as an I/O port in a system which employs the 8-bit parallel processing MSM80C85A CPU. This device has 24-bit I/O pins equivalent to three 8-bit I/O ports and all inputs/outputs are TTL interface compatible.

### FEATURES

- High speed and low power consumption due to 3$\mu$ silicon gate CMOS technology
- 3V to 6V single power supply
- Full static operation
- Programmable 24-bit I/O ports
- Bidirectional bus operation (Port A)
- Bit set/reset function (Port C)
- TTL compatible
- Compatible with 8255A-5
- 40 pin Plastic DIP (DIP40-P-600): MSM82C55A-2RS
- 44 pin Plastic QFJ (QFJ44-P-S650): MSM82C55A-2JS
- 44 pin Plastic QFP (QFP44-P-910-2K): MSM82C55A-2GS-2K

### CIRCUIT CONFIGURATION

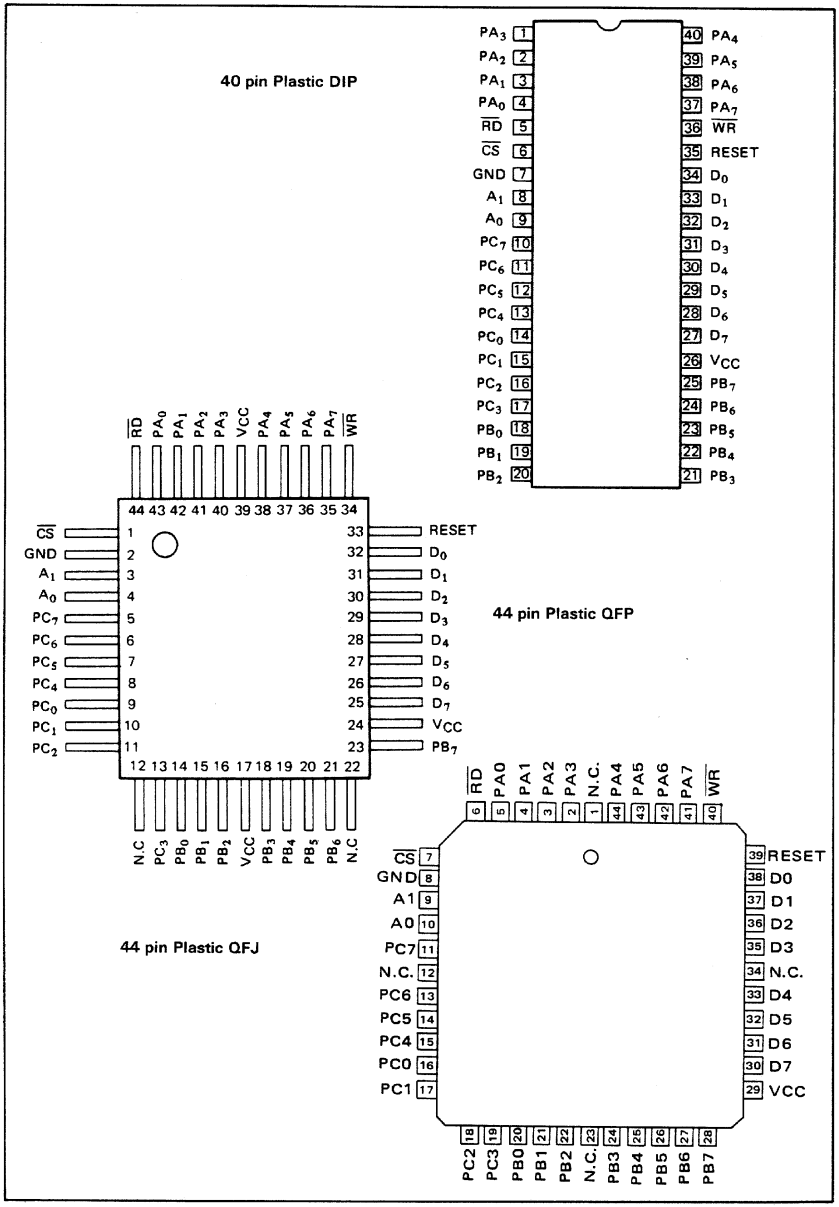





 © National Instruments Corporation

■ I/O-MSM82C55A-2RS/GS/VJS ■

## PIN CONFIGURATION (Top View)



## ■ I/O-MSM82C55A-2RS/GS/VJS ■

### ABSOLUTE MAXIMUM RATINGS

| Parameter | Symbol | Conditions | Limits | | | Unit |
|---|---|---|---|---|---|---|
| | | | MSM82C55A-2RS | MSM82C55A-2GS | MSM82C55A-2VJS | |
| Ssupply Voltage | $V_{CC}$ | Ta = 25°C with respect to GND | −0.5 to +7 | | | V |
| Input Voltage | $V_{IN}$ | | −0.5 to $V_{CC}$ + 0.5 | | | V |
| Output Voltage | $V_{OUT}$ | | −0.5 to $V_{CC}$ + 0.5 | | | V |
| Storage Temperature | $T_{stg}$ | — | −55 to +150 | | | °C |
| Power Dissipation | $P_D$ | Ta = 25°C | 1.0 | 0.7 | 1.0 | W |

### OPERATING RANGE

| Parameter | Symbol | Limits | Unit |
|---|---|---|---|
| Supply Voltage | $V_{CC}$ | 3 to 6 | V |
| Operating Temperature | $T_{OP}$ | −40 to 85 | °C |

### RECOMMENDED OPERATING RANGE

| Parameter | Symbol | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|
| Supply Voltage | $V_{CC}$ | 4.5 | 5 | 5.5 | V |
| Operating Temperature | $T_{OP}$ | −40 | +25 | +85 | °C |
| "L" Input Voltage | $V_{IL}$ | -0.3 | | +0.8 | V |
| "H" Input Voltage | $V_{IH}$ | 2.2 | | $V_{CC}$ +0.3 | V |

### DC CHARACTERISTICS

| Parameter | Symbol | Conditions | | MSM82C55A-2 | | | Unit |
|---|---|---|---|---|---|---|---|
| | | | | Min. | Typ. | Max. | |
| "L" Output Voltage | $V_{OL}$ | $I_{OL}$ = 2.5 mA | $V_{CC}$ = 4.5V to 5.5V Ta = −40°C to +85°C ($C_L$ = 0pF) | | | 0.4 | V |
| "H" Output Voltage | $V_{OH}$ | $I_{OH}$ = −40 μA | | 4.2 | | | V |
| | | $I_{OH}$ = −2.5 mA | | 3.7 | | | V |
| Input Leak Current | $I_{LI}$ | $0 \leq V_{IN} \leq V_{CC}$ | | -1 | | 1 | μA |
| Output Leak Current | $I_{LO}$ | $0 \leq V_{OUT} \leq V_{CC}$ | | -10 | | 10 | μA |
| Supply Current (standby) | $I_{CCS}$ | $\overline{CS} \geqq V_{CC}$ -0.2V<br>$V_{IH} \geqq V_{CC}$ -0.2V<br>$V_{IL} \leqq$ 0.2V | | | 0.1 | 10 | μA |
| Average Supply Current (active) | $I_{CC}$ | I/O wire cycle 82C55A-2 ... 8MHzCPU timing | | | | 8 | mA |

## AC CHARACTERISTICS

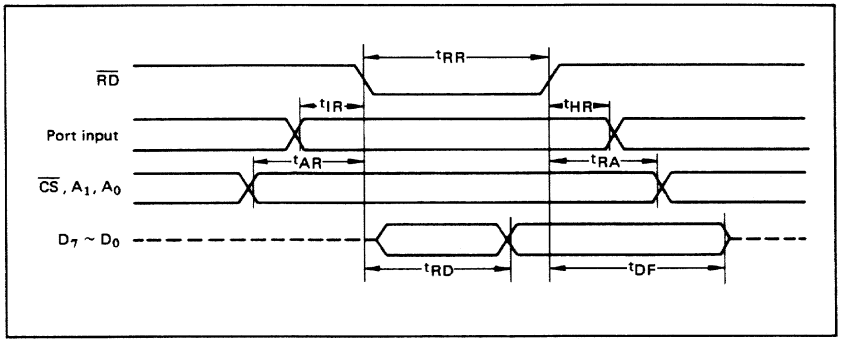
($V_{CC}$ = 4.5 to 5.5V, Ta = -40 to +80°C)

| Parameter | Symbol | MSM82C55A-2 | | Unit | Remarks |
|---|---|---|---|---|---|
| | | Min. | Max. | | |
| Setup Time of address to the falling edge of $\overline{RD}$ | $t_{AR}$ | 20 | | ns | |
| Hold Time of address to the rising edge of $\overline{RD}$ | $t_{RA}$ | 0 | | ns | |
| $\overline{RD}$ Pulse Width | $t_{RR}$ | 100 | | ns | |
| Delay Time from the falling edge of $\overline{RD}$ to the output of defined data | $t_{RD}$ | | 120 | ns | |
| Delay Time from the rising edge of $\overline{RD}$ to the floating of data bus | $t_{DF}$ | 10 | 75 | ns | |
| Time from the rising edge of $\overline{RD}$ or $\overline{WR}$ to the next falling edge of $\overline{RD}$ or $\overline{WR}$ | $t_{RV}$ | 200 | | ns | |
| Setup Time of address before the falling edge of $\overline{WR}$ | $t_{AW}$ | 0 | | ns | |
| Hold Time of address after the rising edge or $\overline{WR}$ | $t_{WA}$ | 20 | | ns | |
| $\overline{WR}$ Pulse Width | $t_{WW}$ | 150 | | ns | |
| Setup Time of bus data before the rising edge of $\overline{WR}$ | $t_{DW}$ | 50 | | ns | |
| Holt Time of bus data after the rising edge of $\overline{WR}$ | $t_{WD}$ | 30 | | ns | |
| Delay Time from the rising edge of $\overline{WR}$ to the output of defined data | $t_{WB}$ | | 200 | ns | |
| Setup Time of port data before the falling edge of $\overline{RD}$ | $t_{IR}$ | 20 | | ns | |
| Hold Time of port data after the rising edge of $\overline{RD}$ | $t_{HR}$ | 10 | | ns | |
| $\overline{ACK}$ Pulse Width | $t_{AK}$ | 100 | | ns | |
| $\overline{STB}$ Pulse Width | $t_{ST}$ | 100 | | ns | Load 150 pF |
| Setup Time of port data before the rising edge of $\overline{STB}$ | $t_{PS}$ | 20 | | ns | |
| Hold Time of port data after the rising edge of $\overline{STB}$ | $t_{PH}$ | 50 | | ns | |
| Delay Time from the falling edge of $\overline{ACK}$ to the output of defined data | $t_{AD}$ | | 150 | ns | |
| Delay Time from the rising edge of $\overline{ACK}$ to the floating of port (Port A in mode 2) | $t_{KD}$ | 20 | 250 | ns | |
| Delay Time from the rising edge of WR to the falling edge of $\overline{OBF}$ | $t_{WOB}$ | | 150 | ns | |
| Delay Time from the falling edge of ACK to the rising edge of $\overline{OBF}$ | $t_{AOB}$ | | 150 | ns | |
| Delay Time from the falling edge of STB to the rising edge of IBF | $t_{SIB}$ | | 150 | ns | |
| Delay Time from the rising edge of $\overline{RD}$ to the falling edge of IBF | $t_{RIB}$ | | 150 | ns | |
| Delay Time from the falling edge of $\overline{RD}$ to the falling edge of INTR | $t_{RIT}$ | | 200 | ns | |
| Delay Time from the rising edge of $\overline{STB}$ to the rising edge of INTR | $t_{SIT}$ | | 150 | ns | |
| Delay Time from the rising edge of $\overline{ACK}$ to the rising edge of INTR | $t_{AIT}$ | | 150 | ns | |
| Delay Time from the falling edge of $\overline{WR}$ to the falling edge of INTR | $t_{WIT}$ | | 250 | ns | |

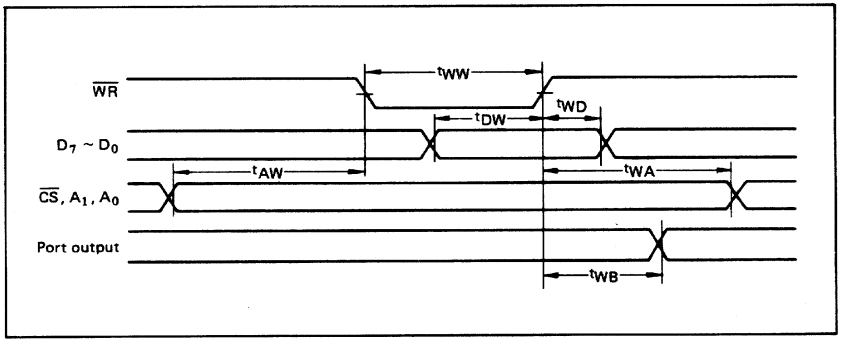
**Note:** Timing is measured at $V_L$ = 0.8 V and $V_H$ = 2.2 V for both input and outputs.

## ■ I/O-MSM82C55A-2RS/GS/VJS ■

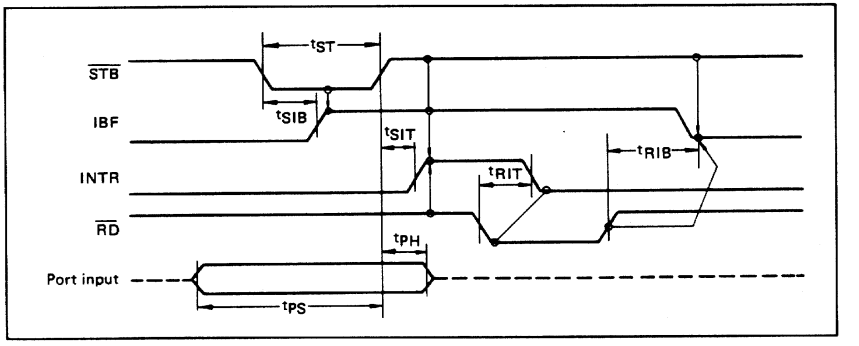
**Basic Input Operation (Mode 0)**




**Basic Output Operation (Mode 0)**

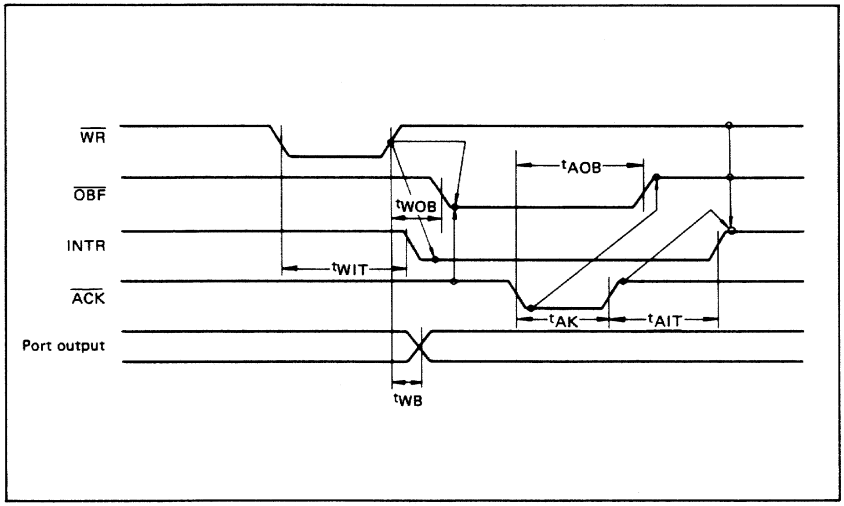



**Strobe Input Operation (Mode 1)**

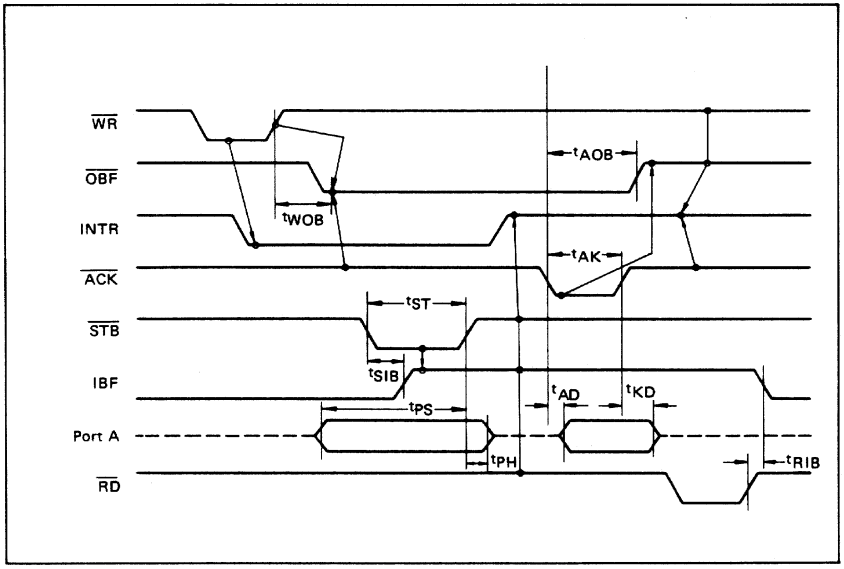



194

**Strobe Output Operation (Mode 1)**



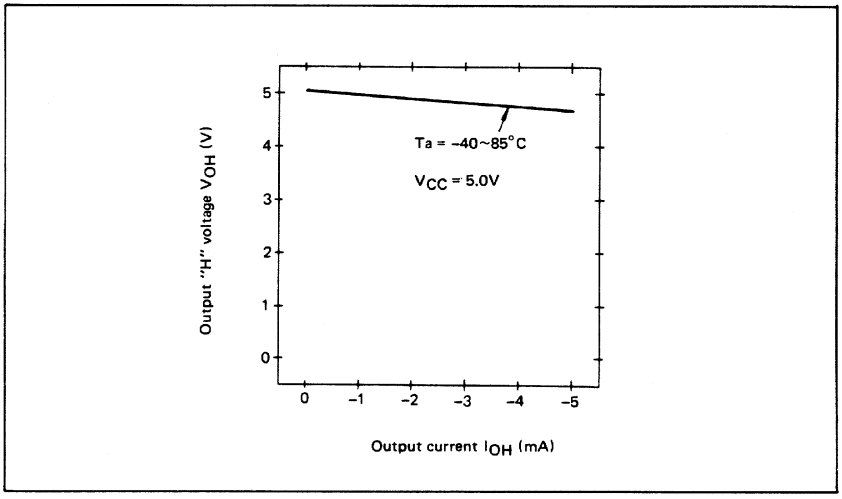

**Bidirectional Bus Operation (Mode 2)**



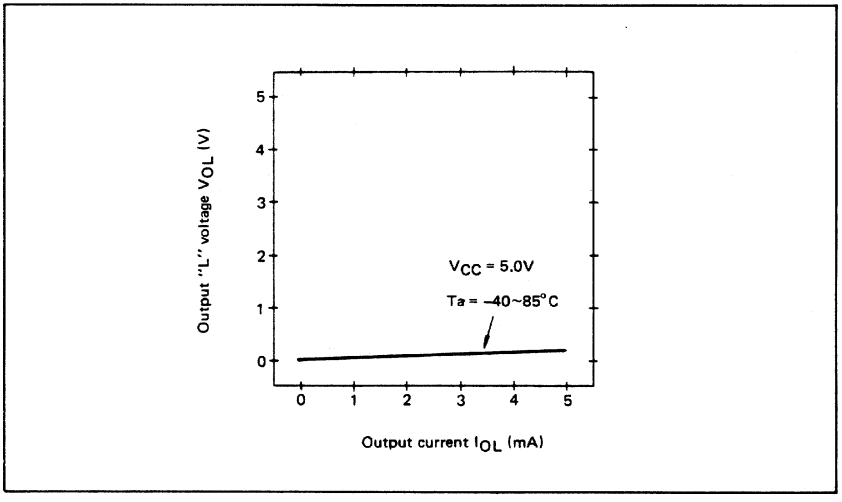
■ I/O-MSM82C55A-2RS/GS/VJS ■

## OUTPUT CHARACTERISTICS (REFERENCE VALUE)

**1 Output "H" Voltage ($V_{OH}$) vs. Output Current ($I_{OH}$)**




**2 Output "L" Voltage ($V_{OL}$) vs. Output Current ($I_{OL}$)**




**Note:** The direction of flowing into the device is taken as positive for the output current.

■ I/O-MSM82C55A-2RS/GS/VJS ■

## FUNCTIONAL DESCRIPTION OF PIN

| Pin No. | Item | Input/Output | Function |
|---|---|---|---|
| D7 ~ D0 | Bidirectional data bus | Input and output | These are three-state 8-bit bidirectional buses used to write and read data upon receipt of the $\overline{WR}$ and $\overline{RD}$ signals from CPU and also used when control words and bit set/reset data are transferred from CPU to MSM82C55A. |
| RESET | Reset input | Input | This signal is used to reset the control register and all internal registers when it is in high level. At this time, ports are all made into the input mode (high impedance status). all port latches are cleared to 0. and all ports groups are set to mode 0. |
| $\overline{CS}$ | Chip select input | Input | When the $\overline{CS}$ is in low level, data transmission is enabled with CPU. When it is in high level, the data bus is made into the high impedance status where no write nor read operation is performed. Internal registers hold their previous status, however. |
| $\overline{RD}$ | Read input | Input | When RD is in low level, data is transferred from MSM82C55A to CPU. |
| $\overline{WR}$ | Write input | Input | When WR is in low level, data or control words are transferred from CPU to MSM82C55A. |
| A0, A1 | Port select input (address) | Input | By combination of A0 and A1, either one is selected from among port A, port B, port C, and control register. These pins are usually connected to low order 2 bits of the address bus. |
| PA7 ~ PA0 | Port A | Input and output | These are universal 8-bit I/O ports. The direction of inputs/outputs can be determined by writing a control word. Especially, port A can be used as a bidirectional port when it is set to mode 2. |
| PB7 ~ PB0 | Port B | Input and output | These are universal 8-bit I/O ports. The direction of inputs/outputs can be determined by writing a control word. |
| PC7 ~ PC0 | Port C | Input and output | These are universal 8-bit I/O ports. The direction of inputs/outputs can be determined by writing a control word as 2 ports with 4 bits each. When port A or port B is used in mode 1 or mode 2 (port A only), they become control pins. Especially when port C is used as an output port, each bit can be set/reset independently. |
| $V_{CC}$ | | | +5 V power supply. |
| GND | | | GND |

## BASIC FUNCTIONAL DESCRIPTION

**Group A and Group B**

When setting a mode to a port having 24 bits, set it by dividing it into two groups of 12 bits each.

Group A: Port A (8 bits) and high order 4 bits of port C (PC7 ~ PC4)

Group B: Port B (8 bits) and low order 4 bits of port C (PC3 ~ PC0)

**Mode 0, 1, 2**

There are 3 types of modes to be set by grouping as follows:

Mode 0: Basic input operation/output operation (Available for both groups A and B)

Mode 1: Strobe input operation/output operation (Available for both groups A and B)

Mode 2: Bidirectional bus operation (Available for group A only)

When used in mode 1 or mode 2, however, port C has bits to be defined as ports for control signal for operation ports (port A for group A and port B for group B) of their respective groups.

**Port A, B, C**

The internal structure of 3 ports is as follows:

Port A: One 8-bit data output latch/buffer and one 8-bit data input latch

Port B: One 8-bit data input/output latch/buffer and one 8-bit data input buffer

Port C: One 8-bit data output latch/buffer and one 8-bit data input buffer (no latch for input)

**Single bit set/reset function for port C**

When port C is defined as an output port, it is possible to set (to turn to high level) or reset (to turn to low level) any one of 8 bits individually without affecting other bits.

■ I/O-MSM82C55A-2RS/GS/VJS ■

## OPERATIONAL DESCRIPTION

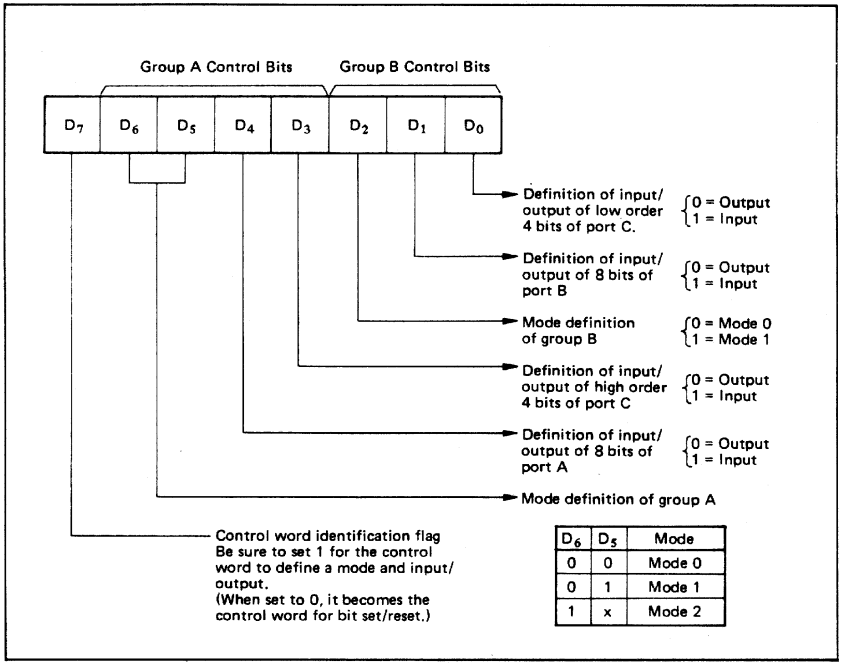
**Control Logic**

Operations by addresses and control signals, e.g., read and write, etc. are as shown in the table below:

| Operation | A1 | A0 | $\overline{CS}$ | $\overline{WR}$ | $\overline{RD}$ | Operation |
|---|---|---|---|---|---|---|
| Input | 0 | 0 | 0 | 1 | 0 | Port A → Data Bus |
| | 0 | 1 | 0 | 1 | 0 | Port B → Data Bus |
| | 1 | 0 | 0 | 1 | 0 | Port C → Data Bus |
| Output | 0 | 0 | 0 | 0 | 1 | Data Bus → Port A |
| | 0 | 1 | 0 | 0 | 1 | Data Bus → Port B |
| | 1 | 0 | 0 | 0 | 1 | Data Bus → Port C |
| Control | 1 | 1 | 0 | 0 | 1 | Data Bus → Control Register |
| Others | 1 | 1 | 0 | 1 | 0 | Illegal Condition |
| | x | x | 1 | x | x | Data bus is in the high impedance status. |

**Setting of Control Word**

The control register is composed of 7-bit latch circuit and 1-bit flag as shown below.


Group A Control Bits
Group B Control Bits
D7 D6 D5 D4 D3 D2 D1 D0
Definition of input/ output of low order 4 bits of port C.
0 = Output
1 = Input
Definition of input/ output of 8 bits of port B
0 = Output
1 = Input
Mode definition of group B
0 = Mode 0
1 = Mode 1
Definition of input/ output of high order 4 bits of port C
0 = Output
1 = Input
Definition of input/ output of 8 bits of port A
0 = Output
1 = Input
Mode definition of group A
Control word identification flag
Be sure to set 1 for the control word to define a mode and input/ output.
(When set to 0, it becomes the control word for bit set/reset.)

| $D_6$ | $D_5$ | Mode |
|---|---|---|
| 0 | 0 | Mode 0 |
| 0 | 1 | Mode 1 |
| 1 | x | Mode 2 |

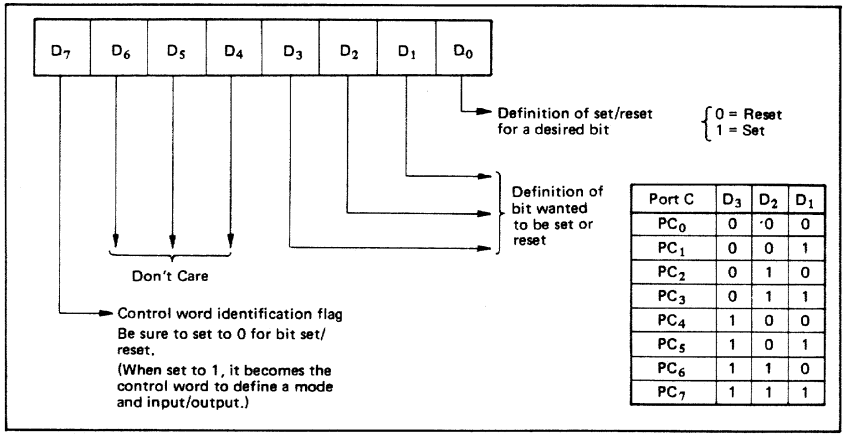
**Precaution for mode selection**

The output registers for ports A and C are cleared to $\phi$ each time data is written in the command register and the mode is changed, but the port B state is undefined.

**Bit Set/Reset Function**

When port C is defined as output port, it is possible to set (set output to 1) or reset (set output to 0) any one of 8 bits without affecting other bits as shown next page.

■ I/O-MSM82C55A-2RS/GS/VJS ■


D7 D6 D5 D4 D3 D2 D1 D0
Definition of set/reset for a desired bit
0 = Reset
1 = Set
Definition of bit wanted to be set or reset
Don't Care
Control word identification flag
Be sure to set to 0 for bit set/ reset.
(When set to 1, it becomes the control word to define a mode and input/output.)

| Port C | $D_3$ | $D_2$ | $D_1$ |
|---|---|---|---|
| $PC_0$ | 0 | 0 | 0 |
| $PC_1$ | 0 | 0 | 1 |
| $PC_2$ | 0 | 1 | 0 |
| $PC_3$ | 0 | 1 | 1 |
| $PC_4$ | 1 | 0 | 0 |
| $PC_5$ | 1 | 0 | 1 |
| $PC_6$ | 1 | 1 | 0 |
| $PC_7$ | 1 | 1 | 1 |

**Interrupt Control Function**

When the MSM82C55A is used in mode 1 or mode 2, the interrupt signal for the CPU is provided. The interrupt request signal is output from port C. When the internal flip-flop INTE is set beforehand at this time, the desired interrupt request signal is output. When it is reset beforehand, however, the interrupt request signal is not output. The set/reset of the internal flip-flop is made by the bit set/reset operation for port C virtually.

Bit set → INTE is set → Interrupt allowed
Bit reset → INTE is reset → Interrupt inhibited

**Operational Description by Mode**

**1. Mode 0 (Basic input/output operation)**

Mode 0 makes the MSM82C55A operate as a basic input port or output port. No control signals such as interrupt request, etc. are required in this mode. All 24 bits can be used as two-8-bit ports and two 4-bit ports. Sixteen combinations are then possible for inputs/outputs. The inputs are not latched, but the outputs are.

| Type | Control Word | | | | | | | | Group A | | Group B | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $D_7$ | $D_6$ | $D_5$ | $D_4$ | $D_3$ | $D_2$ | $D_1$ | $D_0$ | Port A | High Order 4 Bits of Port C | Port B | Low Order 4 Bits of Port C |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | Output | Output | Output | Output |
| 2 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | Output | Output | Output | Input |
| 3 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | Output | Output | Input | Output |
| 4 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | Output | Output | Input | Input |
| 5 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | Output | Input | Output | Output |
| 6 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | Output | Input | Output | Input |
| 7 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | Output | Input | Input | Output |
| 8 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | Output | Input | Input | Input |
| 9 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | Input | Output | Output | Output |
| 10 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | Input | Output | Output | Input |
| 11 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | Input | Output | Input | Output |
| 12 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | Input | Output | Input | Input |
| 13 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | Input | Input | Output | Output |
| 14 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | Input | Input | Output | Input |
| 15 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | Input | Input | Input | Output |
| 16 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | Input | Input | Input | Input |

**Note:** When used in mode 0 for both groups A and B

## ■ I/O-MSM82C55A-2RS/GS/VJS ■

**2. Mode 1 (Strobe input/output operation)**

In mode 1, the strobe, interrupt and other control signals are used when input/output operations are made from a specified port. This mode is available for both groups A and B. In group A at this time, port A is used as the data line and port C as the control signal.

Following is a descrption of the input operation in mode 1.

**$\overline{STB}$ (Strobe input)**

- When this signal is low level, the data output from terminal to port is fetched into the internal latch of the port. This can be made independent from the CPU, and the data is not output to the data bus until the RD signal arrives from the CPU.

**IBF (Input buffer full flag output)**

- This is the response signal for the $\overline{STB}$. This signal when turned to high level indicates that data is fetched into the input latch. This signal turns to high level at the falling edge of $\overline{STB}$ and to low level at the rising edge of $\overline{RD}$.

**INTR (Interrupt request output)**

- This is the interrupt request signal for the CPU of the data fetched into the input latch. It is indicated by high level only when the internal INTE flip-flop is set. This signal turns to high level at the rising edge of the $\overline{STB}$ (IBF = 1 at this time) and low level at the falling edge of the $\overline{RD}$ when the INTE is set.

  $INTE_A$ of group A is set when the bit for $PC_4$ is set, while $INTE_B$ of group B is set when the bit for $PC_2$ is set.

Following is a description of the output operation of mode 1.

**$\overline{OBF}$ (Output buffer full flag output)**

- This signal when turned to low level indicates that data is written to the specified port upon receipt of the $\overline{WR}$ signal from the CPU. This signal turns to low level at the rising edge of the WR and high level at the falling edge of the ACK.

**$\overline{ACK}$ (Acknowledge input)**

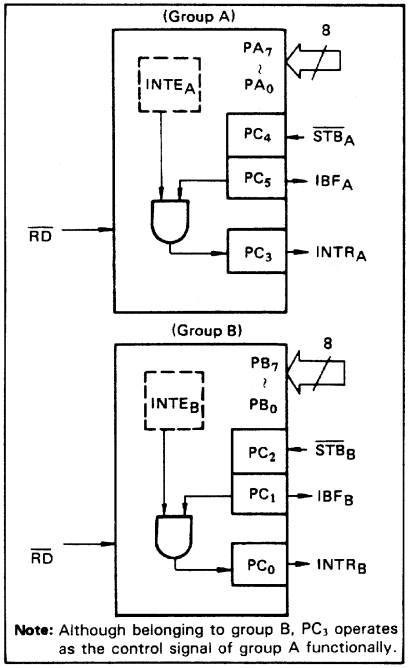
- This signal when turned to low level indicates that the terminal has received data.

**INTR (Interrupt request output)**

- This is the signal used to interrupt the CPU when a terminal receives data from the CPU via the MSM82C55A-5. It indicates the occurrence of the interrupt in high level only when the internal INTE flip-flop is set. This signal turns to high level at the rising edge of the $\overline{ACK}$ (OBF = 1 at this time) and low level at the falling edge of $\overline{WR}$ when the $INTE_B$ is set.

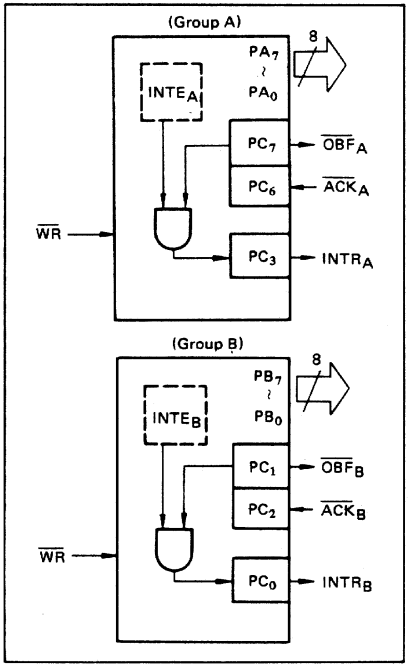
  $INTE_A$ of group A is set when the bit for $PC_6$ is set, while $INTE_B$ of group B is set when the bit for $PC_2$ is set.

**Mode 1 Input**




**Note:** Although belonging to group B, $PC_3$ operates as the control signal of group A functionally.

**Mode 1 output**

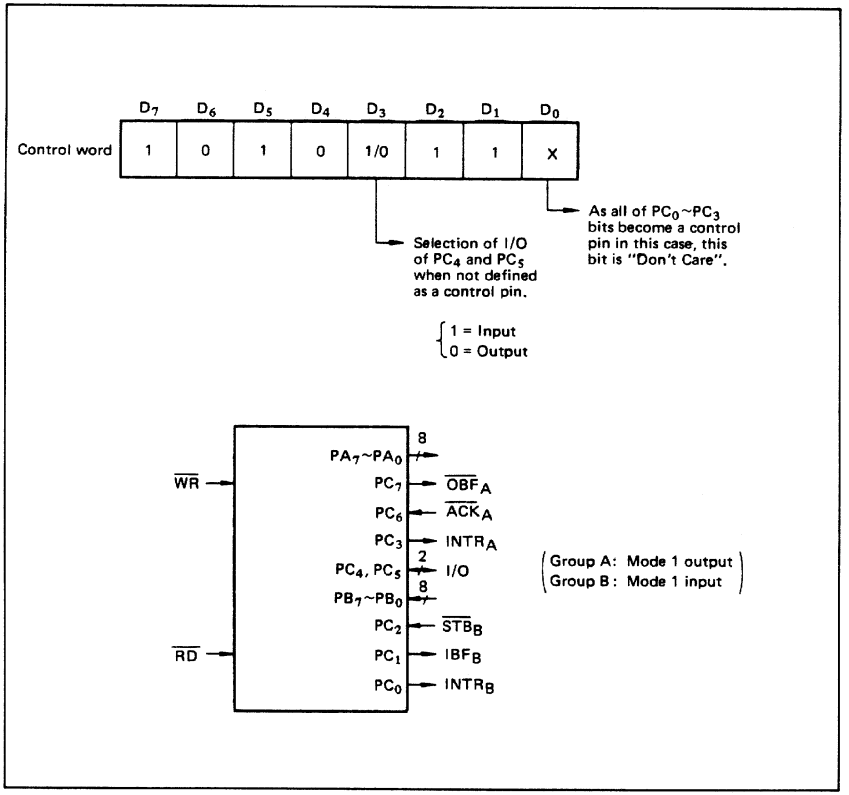


**Port C Function Allocation in Mode 1**

| Combination of Input/Output / Port C | Group A: Input Group B: Input | Group A: Input Group B: Output | Group A: Output Group B: Input | Group A: Output Group B: Output |
|---|---|---|---|---|
| $PC_0$ | $INTR_B$ | $INTR_B$ | $INTR_B$ | $INTR_B$ |
| $PC_1$ | $IBF_B$ | $\overline{OBF}_B$ | $IBF_B$ | $\overline{OBF}_B$ |
| $PC_2$ | $\overline{STB}_B$ | $\overline{ACK}_B$ | $\overline{STB}_B$ | $\overline{ACK}_B$ |
| $PC_3$ | $INTR_A$ | $INTR_A$ | $INTR_A$ | $INTR_A$ |
| $PC_4$ | $\overline{STB}_A$ | $\overline{STB}_A$ | I/O | I/O |
| $PC_5$ | $IBF_A$ | $IBF_A$ | I/O | I/O |
| $PC_6$ | I/O | I/O | $\overline{ACK}_A$ | $\overline{ACK}_A$ |
| $PC_7$ | I/O | I/O | $\overline{OBF}_A$ | $\overline{OBF}_A$ |

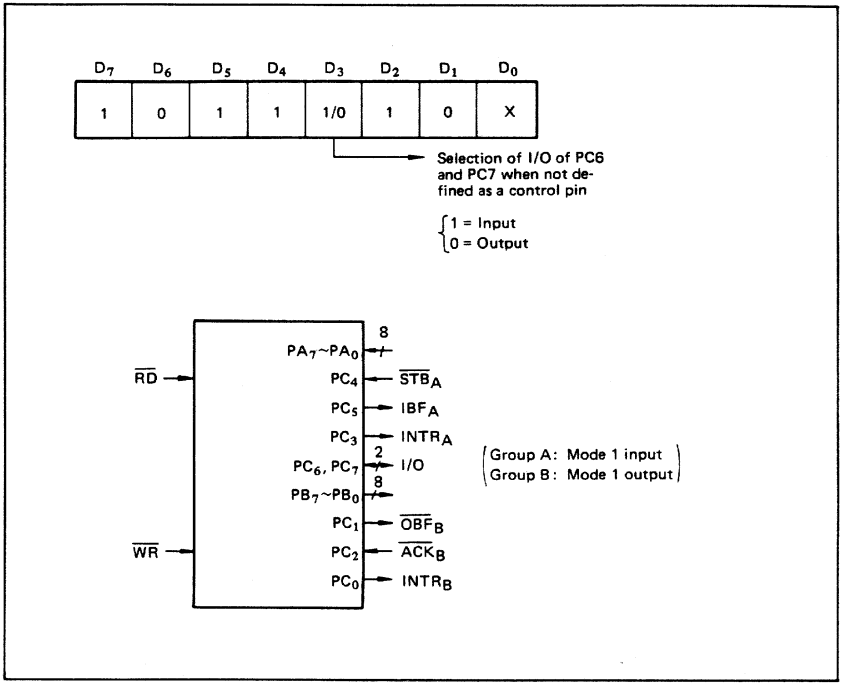
**Note:** I/O is a bit not used as the control signal, but it is available as a port of mode 0.

Examples of the relation between the control words and pins when used in mode 1 is shown below:
(a) When group A is mode 1 output and group B is mode 1 input.


D7 D6 D5 D4 D3 D2 D1 D0
Control word
1 0 1 0 1/0 1 1 X
As all of PC0~PC3 bits become a control pin in this case, this bit is "Don't Care".
Selection of I/O of PC4 and PC5 when not defined as a control pin.
1 = Input
0 = Output
WR
RD
PA7~PA0
PB7~PB0
8
8
PC7 OBFA
PC6 ACKA
PC3 INTRA
PC4, PC5 2 I/O
PC2 STBB
PC1 IBFB
PC0 INTRB
Group A: Mode 1 output
Group B: Mode 1 input

## ■ I/O-MSM82C55A-2RS/GS/VJS ■

(b) When group A is mode 1 input and group B is mode 1 output.




**3. Mode 2 (Strobe bidirectional bus I/O operation)**

In mode 2, it is possible to transfer data in 2 directions through a single 8-bit port. This operation is akin to a combination between input and output operations. Port C waits for the control signal in this case, too. Mode 2 is available only for group A, however.

Next, a description is made on mode 2.

**$\overline{OBF}$ (Output buffer full flag output)**

- This signal when turned to low level indicates that data has been written to the internal output latch upon receipt of the $\overline{WR}$ signal from the CPU. At this time, port A is still in the high impedance status and the data is not yet output to the outside. This signal turns to low level at the rising edge of the $\overline{WR}$ and high level at the falling edge of the $\overline{ACK}$.

**$\overline{ACK}$ (Acknowledge input)**

- When a low level signal is input to this pin, the high impedance status of port A is cleared, the buffer is enabled, and the data written to the internal output latch is output to port A. When the input returns to high level, port A is made into the high impedance status.

**$\overline{STB}$ (Strobe input)**

- When this signal turns to low level, the data output to the port from the pin is fetched into the internal input latch. The data is output to the data bus upon receipt of the RD signal from the CPU, but it remains in the high impedance status until then.

**IBF (Input buffer full flag output)**

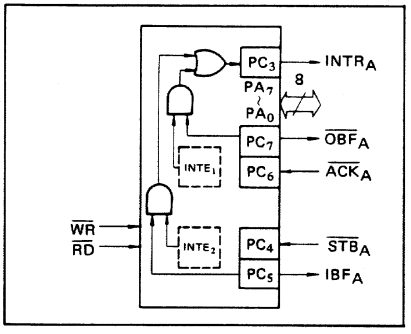
- This signal when turned to high level indicates that data from the pin has been fetched into the input latch. This signal turns to high level at the falling edge of the $\overline{STB}$ and low level at the rising edge of the $\overline{RD}$.

**INTR (Interrupt request output)**

- This signal is used to interrupt the CPU and its operation in the same as in mode 1. There are two INTE flip-flops internally available for input and output to select either interrupt of input or output operation. The INTE1 is used to control the interrupt request for output operation and it can be reset by the bit set for PC6. INTE2 is used to control the interrupt request for the input operation and it can be set by the bit set for PC4.

■ I/O-MSM82C55A-2RS/GS/VJS ■

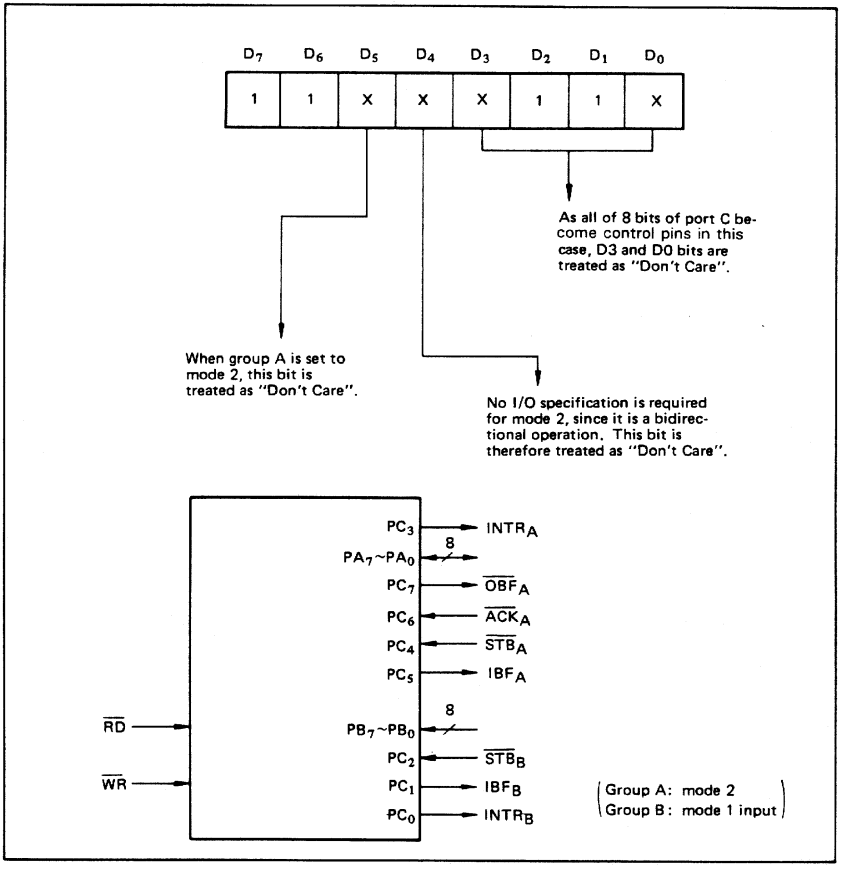
**Mode 2 I/O Operation**




**Port C Function Allocation in Mode 2**

| Port C | Function |
|---|---|
| $PC_0$ | Confirmed to the group B mode |
| $PC_1$ | |
| $PC_2$ | |
| $PC_3$ | $INTR_A$ |
| $PC_4$ | $\overline{STB}_A$ |
| $PC_5$ | $IBF_A$ |
| $PC_6$ | $\overline{ACK}_A$ |
| $PC_7$ | $\overline{OBF}_A$ |

Following is an example of the relation between the control word and the pin when used in mode 2. When input in mode 2 for group A and in mode 1 for group B.



## ■ I/O-MSM82C55A-2RS/GS/VJS ■

**4. When Group A is Different in Mode from Group B**

Group A and group B can be used by setting them in different modes each other at the same time. When either group is set to mode1 or mode 2, it is possible to set the one not defined as a control pin in port C to both input and output as a port which operates in mode 0 at the 3rd and 0th bits of the control word.

**(Mode combinations that define no control bit at port C)**

| | Group A | Group B | Port C | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | $PC_7$ | $PC_6$ | $PC_5$ | $PC_4$ | $PC_3$ | $PC_2$ | $PC_1$ | $PC_0$ |
| 1 | Mode 1 input | Mode 0 | I/O | I/O | $IBF_A$ | $\overline{STB}_A$ | $INTR_A$ | I/O | I/O | I/O |
| 2 | Mode 0 output | Mode 0 | $\overline{OBF}_A$ | $\overline{ACK}_A$ | I/O | I/O | $INTR_A$ | I/O | I/O | I/O |
| 3 | Mode 0 | Mode 1 input | I/O | I/O | I/O | I/O | I/O | $\overline{STB}_B$ | $IBF_B$ | $INTR_B$ |
| 4 | Mode 0 | Mode 1 output | I/O | I/O | I/O | I/O | I/O | $\overline{ACK}_B$ | $\overline{OBF}_B$ | $INTR_B$ |
| 5 | Mode 1 input | Mode 1 input | I/O | I/O | $IBF_A$ | $\overline{STB}_A$ | $INTR_A$ | $\overline{STB}_B$ | $IBF_B$ | $INTR_B$ |
| 6 | Mode 1 input | Mode 1 output | I/O | I/O | $IBF_A$ | $\overline{STB}_A$ | $INTR_A$ | $\overline{ACK}_B$ | $\overline{OBF}_B$ | $INTR_B$ |
| 7 | Mode 1 output | Mode 1 input | $\overline{OBF}_A$ | $\overline{ACK}_A$ | I/O | I/O | $INTR_A$ | $\overline{STB}_B$ | $IBF_B$ | $INTR_B$ |
| 8 | Mode 1 output | Mode 1 output | $\overline{OBF}_A$ | $\overline{ACK}_A$ | I/O | I/O | $INTR_A$ | $\overline{ACK}_B$ | $\overline{OBF}_B$ | $INTR_B$ |
| 9 | Mode 2 | Mode 0 | $\overline{OBF}_A$ | $\overline{ACK}_A$ | $IBF_A$ | $\overline{STB}_A$ | $INTR_A$ | I/O | I/O | I/O |
| | | | Controlled at the 3rd bit (D3) of the control word | | | | Controlled at the 0th bit (D0) of the control word | | | |

When the I/O bit is set to input in this case, it is possible to access data by the normal port C read operation.
When set to output, PC7 ~ PC4 bits can be accessed by the bit set/reset function only. Meanwhile, 3 bits from PC2 to PC0 can be accessed by normal write operation.

The bit set/reset function can be used for all of PC3 ~ PC0 bits. Note that the status of port C varies according to the combination of modes like this.

■ I/O-MSM82C55A-2RS/GS/VJS ■

**5. Port C Status Read**

When port C is used for the control signal, that is, in either mode 1 or mode 2, each control signal and bus status signal can be read out by reading the content of port C.
The status read out is as follows:

| | Group A | Group B | Status read on the data bus | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | $D_7$ | $D_6$ | $D_5$ | $D_4$ | $D_3$ | $D_2$ | $D_1$ | $D_0$ |
| 1 | Mode 1 input | Mode 0 | I/O | I/O | $IBF_A$ | $INTE_A$ | $INTR_A$ | I/O | I/O | I/O |
| 2 | Mode 1 output | Mode 0 | $\overline{OBF}_A$ | $INTE_A$ | I/O | I/O | $INTR_A$ | I/O | I/O | I/O |
| 3 | Mode 0 | Mode 1 input | I/O | I/O | I/O | I/O | I/O | $INTE_B$ | $IBF_B$ | $INTR_B$ |
| 4 | Mode 0 | Mode 1 output | I/O | I/O | I/O | I/O | I/O | $INTE_B$ | $\overline{OBF}_B$ | $INTR_B$ |
| 5 | Mode 1 input | Mode 1 input | I/O | I/O | $IBF_A$ | $INTE_A$ | $INTR_A$ | $INTE_B$ | $IBF_B$ | $INTR_B$ |
| 6 | Mode 1 input | Mode 1 output | I/O | I/O | $IBF_A$ | $INTE_A$ | $INTR_A$ | $INTE_B$ | $\overline{OBF}_B$ | $INTR_B$ |
| 7 | Mode 1 output | Mode 1 input | $\overline{OBF}_A$ | $INTE_A$ | I/O | I/O | $INTR_A$ | $INTE_B$ | $IBF_B$ | $INTR_B$ |
| 8 | Mode 1 output | Mode 1 output | $\overline{OBF}_A$ | $INTE_A$ | I/O | I/O | $INTR_A$ | $INTE_B$ | $\overline{OBF}_B$ | $INTR_B$ |
| 9 | Mode 2 | Mode 0 | $\overline{OBF}_A$ | $INTE_1$ | $IBF_A$ | $INTE_2$ | $INTR_A$ | I/O | I/O | I/O |
| 10 | Mode 2 | Mode 1 input | $\overline{OBF}_A$ | $INTE_1$ | $IBF_A$ | $INTE_2$ | $INTR_A$ | $INTE_B$ | $IBF_B$ | $INTR_B$ |
| 11 | Mode 2 | Mode 1 output | $\overline{OBF}_A$ | $INTE_1$ | $IBF_A$ | $INTE_2$ | $INTR_A$ | $INTE_B$ | $\overline{OBF}_B$ | $INTR_B$ |

**6. Reset of MSM82C55A**

Be sure to keep the RESET signal at power ON in the high level at least for 50 μs. Subsequently, it becomes the input mode at a high level pulse above 500 ns.

**Note:** Comparison of MSM82C55A-5 and MSM82C55A-2

**MSM82C55A-5**

After a write command is executed to the command register, the internal latch is cleared in PORTA PORTC. For instance, 00H is output at the beginning of a write command when the output port is assigned. However, if PORTB is not cleared at this time, PORTB is unstable. In other words, PORTB only outputs ineffective data (unstable value according to the device) during the period from after a write command is executed till the first data is written to PORTB.

**MSM82C55A-2**

After a write command is executed to the command register, the internal latch is cleared in All Ports(PORTA,PORTB,PORTC). 00H is ontput at the beginning of a write command when the output port is assigned.

# D

# MSM82C53 Data Sheet[1]

This appendix contains a manufacturer data sheet for the MSM82C53 CMOS programmable interval timer from OKI Semiconductor. This timer is used on the PCI-DIO-96 and PXI-6508.

[1] Copyright © OKI Semiconductor. 1995. Reprinted with permission of copyright owner. All rights reserved. OKI Semiconductor. *Microprocessor Data Book 1995*.

**OKI semiconductor**

# MSM82C53-2RS/GS/JS

## CMOS PROGRAMMABLE INTERVAL TIMER

### GENERAL DESCRIPTION

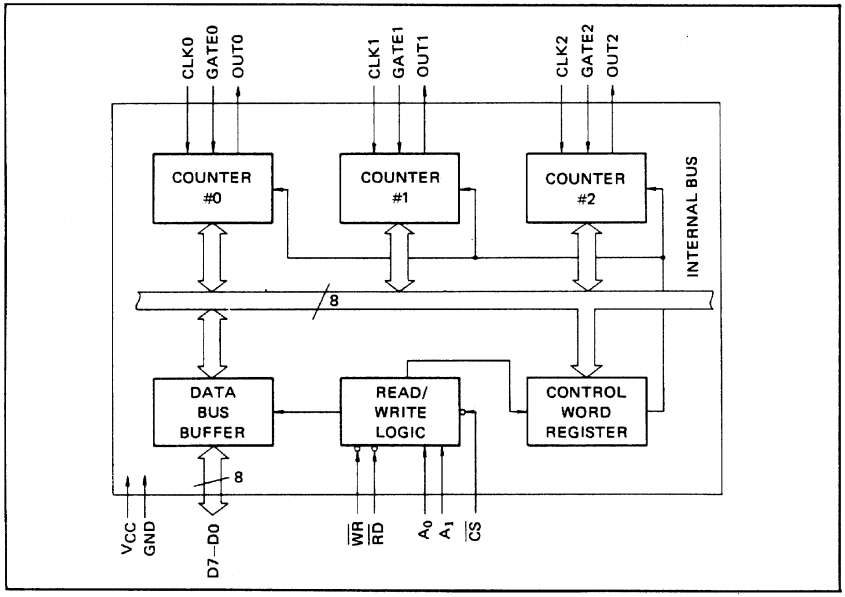
The MSM82C53-2RS/GS/JS is programmable universal timers designed for use in microcomputer systems. Based on silicon gate CMOS technology, it requires a standby current of only 100 μA (max.) when the chip is in the nonselected state. During timer operation, power consumption is still very low with only 8 mA (max.) at 8 MHz of current required.

The device consists of three independent counters, and can count up to a maximum of 8 MHz (MSM82C53-2) The timer features six different counter modes, and binary count/BCD count functions. Count values can be set in byte or word units, and all functions are freely programmable.

### FEATURES

- Maximum operating frequency of 8 MHz (MSM82C53-2)
- High speed and low power consumption achieved through silicon gate CMOS technology
- Completely static operation
- Three independent 16-bit down-counters
- 3V to 6V single power supply
- Six counter modes abailable for each counter
- Binary and decimal counting possible
- 24 pin Plastic DIP (DIP24-P-600): MSM82C53-2RS
- 28 pin Plastic QFJ (QFJ28-P-S450): MSM82C53-2JS
- 32 pin-V Plastic SOP (SSOP32-P-430-VK): MSM82C53-2GS-VK

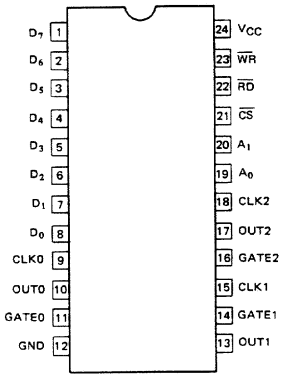
### FUNCTIONAL BLOCK DIAGRAM



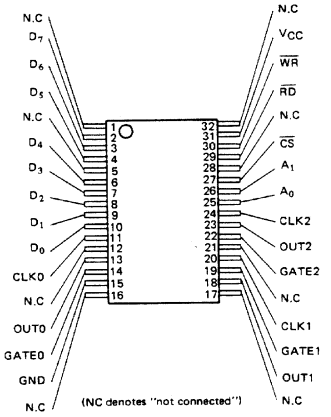
■ I/O-MSM82C53-2RS/GS/JS ■

## PIN CONFIGURATION (Top View)

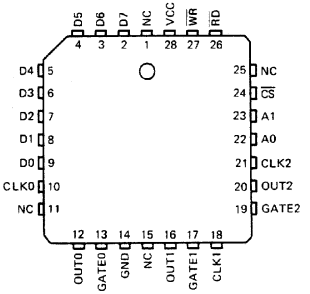
24 pin Plastic DIP




32 pin Plastic SOP




28 pin Plastic QFJ



## ABSOLUTE MAXIMUM RATINGS

| Parameter | Symbol | Conditions | Limits | | | Unit |
|---|---|---|---|---|---|---|
| | | | MSM82C53-2RS | MSM82C53-2GS | MSM82C53-2JS | |
| Ssupply Voltage | $V_{CC}$ | Respect to GND | −0.5 to +7 | | | V |
| Input Voltage | $V_{IN}$ | | −0.5 to $V_{CC}$ + 0.5 | | | V |
| Output Voltage | $V_{OUT}$ | | −0.5 to $V_{CC}$ + 0.5 | | | V |
| Storage Temperature | $T_{stg}$ | | −55 to +150 | | | °C |
| Power Dissipation | $P_D$ | Ta = 25°C | 0.9 | 0.7 | 0.9 | W |

## OPERATING RANGES

| Parameter | Symbol | Limits | Conditions | Unit |
|---|---|---|---|---|
| Supply Voltage | $V_{CC}$ | 3 to 6 | $V_{IL}$ = 0.2V, $V_{IH}$ = $V_{CC}$ − 0.2V, operating frequency 2.6 MHz | V |
| Operating Temperature | $T_{OP}$ | −40 to +85 | | °C |

## RECOMMENDED OPERATING CONDITIONS

| Parameter | Symbol | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|
| Supply Voltage | $V_{CC}$ | 4.5 | 5 | 5.5 | V |
| Operating Temperature | $T_{OP}$ | −40 | +25 | +85 | °C |
| "L" Input Voltage | $V_{IL}$ | −0.3 | | +0.8 | V |
| "H" Input Voltage | $V_{IH}$ | 2.2 | | $V_{CC}$ + 0.3 | V |

## DC CHARACTERISTICS

| Parameter | Symbol | Conditions | | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|---|
| "L" Output Voltage | $V_{OL}$ | $I_{OL}$ = 4mA | | | | 0.45 | V |
| "H" Output Voltage | $V_{OH}$ | $I_{OH}$ = −1mA | | 3.7 | | | V |
| Input Leak Current | $I_{LI}$ | $0 \leq V_{IN} \leq V_{CC}$ | $V_{CC}$=4.5V to 5.5V | −10 | | 10 | μA |
| Output Leak Current | $I_{LO}$ | $0 \leq V_{OUT} \leq V_{CC}$ | Ta=−40°C to +85°C | −10 | | 10 | μA |
| Standby Supply Current | $I_{CCS}$ | $\overline{CS} \geqq V_{CC}$ −0.2V<br>$V_{IH} \geqq V_{CC}$ − 0.2V<br>$V_{IL} \leqq 0.2V$ | | | | 100 | μA |
| Operating Supply Current | $I_{CC}$ | $t_{CLK}$ = 125 ns $C_L$ = 0pF | | | | 8 | mA |

■ I/O-MSM82C53-2RS/GS/JS ■

## AC CHARACTERISTICS

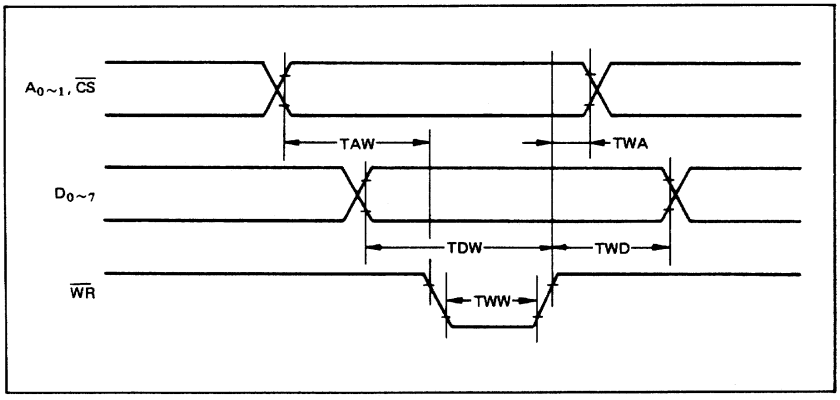
($V_{CC}$ = 4.5V ~ 5.5V, Ta = –40 ~ +85°C)

| Parameter | Symbol | MSM82C53-2 Min. | MSM82C53-2 Max. | Unit | Conditions | |
|---|---|---|---|---|---|---|
| Address Set-up Time before reading | TAR | 30 | | ns | | $C_L$ = 150pF |
| Address Hold Time after reading | TRA | 0 | | ns | Read cycle | |
| Read Pulse Width | TRR | 150 | | ns | | |
| Read Recovery Time | TRVR | 200 | | ns | | |
| Address Set-up Time before writing | TAW | 0 | | ns | | |
| Address Hold Time after writing | TWA | 20 | | ns | | |
| Write Pulse Width | TWW | 150 | | ns | Write cycle | |
| Data Input Set-up Time before writing | TDW | 100 | | ns | | |
| Data Input Hold Time after writing | TWD | 20 | | ns | | |
| Write Recovery time | TRVW | 200 | | ns | | |
| Clock Cycle Time | TCLK | 125 | D.C. | ns | | |
| Clock "H" Pulse Width | TPWH | 60 | | ns | | |
| Clock "L" Pulse Width | TPWL | 60 | | ns | Clock and gate timing | |
| "H" Gate Pulse Width | TGW | 50 | | ns | | |
| "L" Gate Pulse Width | TGL | 50 | | ns | | |
| Gate Input Set-up Time before clock | TGS | 50 | | ns | | |
| Gate Input Hold Time after clock | TGH | 50 | | ns | | |
| Output Delay Time after reading | TRD | | 120 | ns | | |
| Output Floating Delay Time after reading | TDF | 5 | 90 | ns | | |
| Output Delay Time after gate | TODG | | 120 | ns | Delay time | |
| Output Delay Time after clock | TOD | | 150 | ns | | |
| Output Delay Time after address | TAD | | 180 | ns | | |

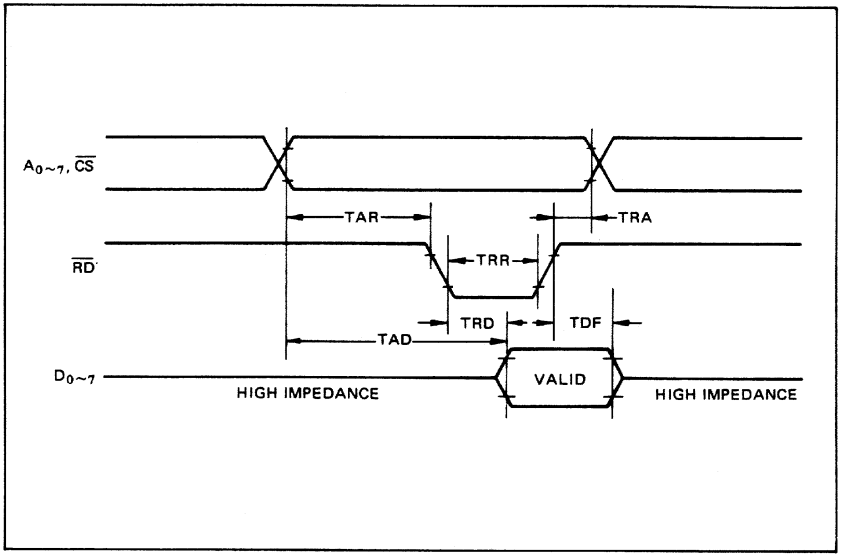
**Note:** Timing measured at $V_L$ = 0.8V and $V_H$ = 2.2V for both inputs and outputs.

## TIME CHART

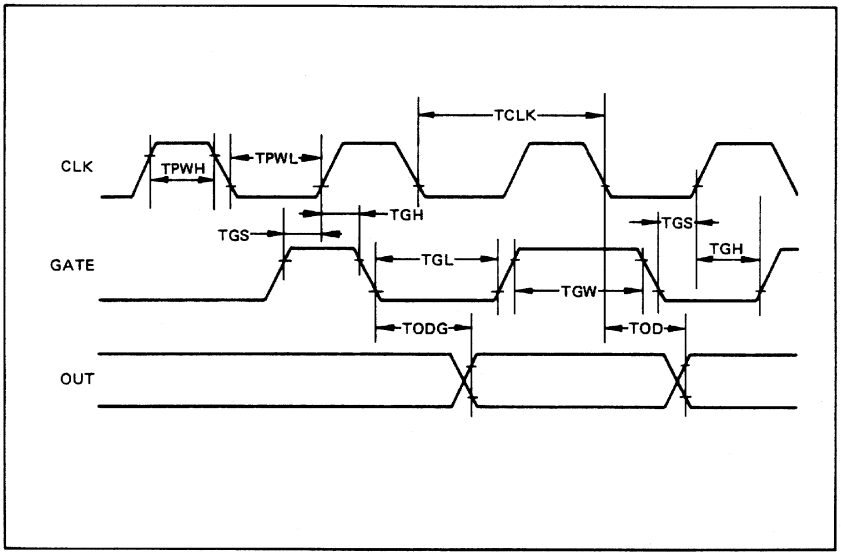
**Write Timing**



**Read Timing**




**Clock & Gate Timing**

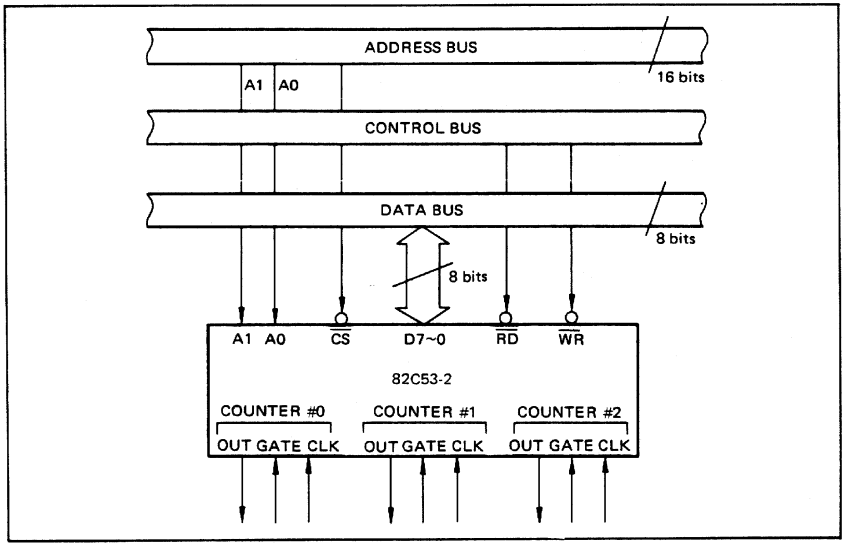


■ I/O-MSM82C53-2RS/GS/JS ■

## DESCRIPTION OF PIN FUNCTIONS

| Pin Symbol | Name | Input/output | Function |
|---|---|---|---|
| D7 ~ D0 | Bidirectional data bus | Input/output | Three-state 8-bit bidirectional data bus used when writing control words and count values, and reading count values upon reception of $\overline{WR}$ and $\overline{RD}$ signals from CPU. |
| $\overline{CS}$ | Chip select input | Input | Data transfer with the CPU is enabled when this pin is at low level. When at high level, the data bus ($D_0$ thru $D_7$) is switched to high impedance state where neither writing nor reading can be executed. Internal registers, however, remain unchanged. |
| $\overline{RD}$ | Read input | Input | Data can be transferred from MSM82C53 to CPU when this pin is at low level. |
| $\overline{WR}$ | Write input | Input | Data can be transferred from CPU to MSM82C53 when this pin is at low level. |
| A0, A1 | Address input | Input | One of the three internal counters or the control word register is selected by A0/A1 combination. These two pins are normally connected to the two lower order bits of the address bus. |
| CLK0~2 | Clock input | Input | Supply of three clock signals to the three counters incorporated in MSM82C53. |
| GATE0~2 | Gate input | Input | Control of starting, interruption, and restarting of counting in the three respective counters in accordance with the set control word contents. |
| OUT0~2 | Counter output | Output | Output of counter output waveform in accordance with the set mode and count value. |

## SYSTEM INTERFACING



## DESCRIPTION OF BASIC OPERATIONS

Data transfers between the internal registers and the external data bus is outlined in the following table.

| $\overline{CS}$ | $\overline{RD}$ | $\overline{WR}$ | A1 | A0 | Function |
|---|---|---|---|---|---|
| 0 | 1 | 0 | 0 | 0 | Data bus to counter # 0 Writing |
| 0 | 1 | 0 | 0 | 1 | Data bus to counter # 1 Writing |
| 0 | 1 | 0 | 1 | 0 | Data bus to counter # 2 Writing |
| 0 | 1 | 0 | 1 | 1 | Data bus to control word register Writing |
| 0 | 0 | 1 | 0 | 0 | Data bus from counter # 0 Reading |
| 0 | 0 | 1 | 0 | 1 | Data bus from counter # 1 Reading |
| 0 | 0 | 1 | 1 | 0 | Data bus from counter # 2 Reading |
| 0 | 0 | 1 | 1 | 1 | Data bus in high impedance status |
| 1 | x | x | x | x | |
| 0 | 1 | 1 | x | x | |

x denotes "not specified".

## DESCRIPTION OF OPERATION

82C53 functions are selected by a control word from the CPU. In the required program sequence, the control word setting is followed by the count value setting and execution of the desired timer operation.

**Control Word and Count Value Program**

Each counter operation mode is set by control word programming. The control word format is outlined below.

| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|
| SC1 | SC0 | RL1 | RL0 | M2 | M1 | M0 | BCD |
| Select Counter | | Read/Load | | Mode | | | BCD |

($\overline{CS}$ = 0, A0, A1 = 1,1, $\overline{RD}$ = 1, $\overline{WR}$ = 0)

- **Select Counter (SC0, SC1):** Selection of set counter

| SC1 | SC0 | Set Contents |
|---|---|---|
| 0 | 0 | Counter # 0 selection |
| 0 | 1 | Counter # 1 selection |
| 1 | 0 | Counter # 2 selection |
| 1 | 1 | Illegal combination |

- **Read/Load (RL1, RL0):** Count value Reading/Loading format setting

| RL1 | RL0 | Set Contents |
|---|---|---|
| 0 | 0 | Counter Latch operation |
| 0 | 1 | Reading/Loading of Least Significant byte (LSB) |
| 1 | 0 | Reading/Loading of Most Significant byte (MSB) |
| 1 | 1 | Reading/Loading of LSB followed by MSB |

- **Mode (M2, M1, M0):** Operation waveform mode setting

| M2 | M1 | M0 | Set Contents |
|---|---|---|---|
| 0 | 0 | 0 | Mode 0 (Interrupt on Terminal Count) |
| 0 | 0 | 1 | Mode 1 (Programmable One-Shot) |
| x | 1 | 0 | Mode 2 (Rate Generator) |
| x | 1 | 1 | Mode 3 (Square Wave Generator) |
| 1 | 0 | 0 | Mode 4 (Software Triggered Strobe) |
| 1 | 0 | 1 | Mode 5 (Hardware Triggered Strobe) |

x denotes "not specified".

- **BCD:** Operation count mode setting

| BCD | Set Contents |
|---|---|
| 0 | Binary Count (16-bits Binary) |
| 1 | BCD Count (4-decades Binary Coded Decimal) |

After setting Read/Load, Mode, and BCD in each counter as outlined above, next set the desired count value. (In some Modes, counting is started immediately after the count value has been written). This count value setting must conform with the Read/Load format set in advance. Note that the internal counters are reset to OOOOH during control word setting. The counter value (OOOOH) can.t be read.

If the two bytes (LSB and MSB) are written at this stage (RL0 and RL1 = 1,1), take note of the following precaution.

Although the count values may be set in the three counters in any sequence after the control word has been set in each counter, count values must be set consecutively in the LSB – MSB order in any one counter.

## ■ I/O-MSM82C53-2RS/GS/JS ■

- **Example of control word and count value setting**

```
Counter #0: Read/Load LSB only, Mode 3,
            Binary count, count value 3H
Counter #1: Read/Load MSB only, Mode 5,
            Binary count, count value AA00H
Counter #2: Read/Load LSB and MSB, Mode 0,
            BCD count, count value 1234

MVI A, 1EH
OUT n3      Counter #0 control word setting
MVI A, 6AH
OUT n3      Counter #1 control word setting
MVI A, B1H
OUT n3      Counter #2 control word setting
MVI A, 03H
OUT n0      Counter #0 count value setting
MVI A, AAH
OUT n1      Counter #1 count value setting
MVI A, 34H
OUT n2      Counter #2 count value setting
MVI A, 12H  (LSB then MSB)
OUT n2
```

**Notes:** n0: Counter #0 address
n1: Counter #1 address
n2: Counter #2 address
n3: Control word register address

- **The minimum and maximum count values which can be counted in each mode are listed below.**

| Mode | Min. | Max. | Remarks |
|---|---|---|---|
| 0 | 1 | 0 | 0 executes 10000H count (ditto in other modes) |
| 1 | 1 | 0 | |
| 2 | 2 | 0 | 1 cannot be counted |
| 3 | 2 | 1 | 1 executes 10001H count |
| 4 | 1 | 0 | |
| 5 | 1 | 0 | |

**Mode Definition**

- **Mode 0 (terminal count)**
  The counter output is set to "L" level by the mode setting. If the count value is then written in the counter with the gate input at "H" level (that is, upon completion of writing the MSB when there are two bytes), the clock input counting is started. When the terminal count is reached, the output is switched to "H" level and is maintained in this status until the control word and count value are set again.
  Counting is interrupted if the gate input is switched to "L" level, and restarted when switched back to "H" level.
  When Count Values are written during counting, the operation is as follows:

  1-byte Read/Load.... When the new count value is written, counting is stopped immediately, and then restarted at the new count value by the next clock.
  2-byte Read/Load.... When byte 1 (LSB) of the new count value is written, counting is stopped immediately. Counting is restarted at the new count value when byte 2 (MSB) is written.

- **Mode 1 (programmable one-shot)**
  The counter output is switched to "H" level by the mode setting. Note that in this mode, counting is not started if only the count value is written. Since counting has to be started in this mode by using the leading edge of the gate input as a trigger, the counter output is switched to "L" level by the next clock after the gate input trigger. This "L" level status is maintained during the set count value, and is switched back to "H" level when the terminal count is reached.
  Once counting has been started, there is no interruption until the terminal count is reached, even if the gate input is switched to "L" level in the meantime. And although counting continues even if a new count value is written during the counting, counting is started at the new count value if another trigger is applied by the gate input.

- **Mode 2 (rate generator)**
  The counter output is switched to "H" level by the mode setting. When the gate input is at "H" level, counting is started by the next clock after the count value has been written. And if the gate input is at "L" level, counting is started by using the rising edge of the gate input as a trigger after the count value has been set.
  An "L" level output pulse appears at the counter output during a single clock duration once every n clock inputs where n is the set count value. If a new count value is written during while counting is in progress, counting is started at the new count value following output of the pulse currently being counted. And if the gate input is switched to "L" level during counting, the counter output is forced to switch to "H" level, the counting being restarted by the rising edge of the gate input.

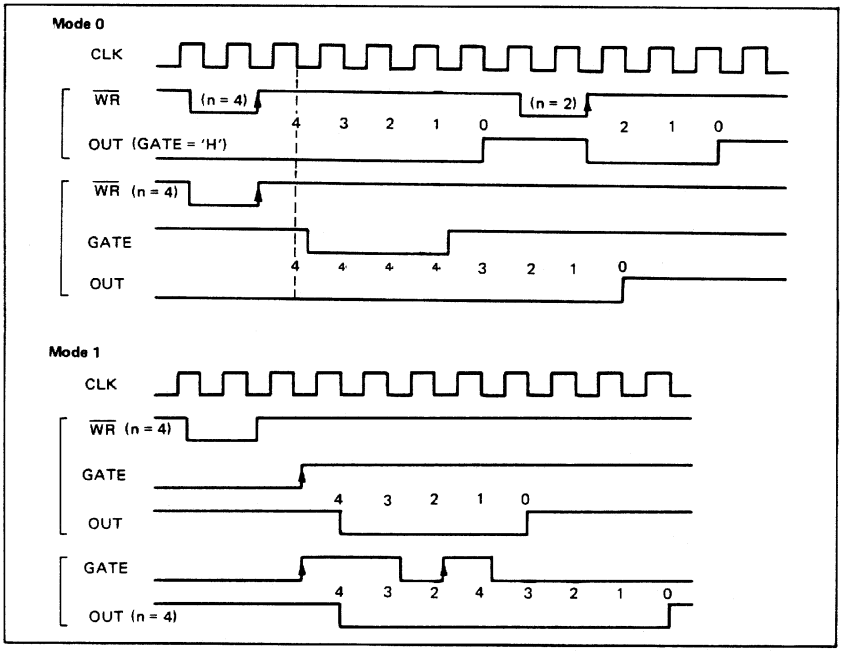
- **Mode 3 (square waveform rate generator)**
  The counter output is switched to "H" level by the mode setting. Counting is started in the same way as described for mode 2 above.
  The repeated square wave output appearing at the counter output contains half the number of counts as the set count value. If the set count value (n) is an odd number, the repeated square wave output consists of only (n + 1)/2 clock inputs at "H" level and (n − 1)/2 clock inputs at "L" level.
  If a new count value is written during counting, the new count value is reflected immediately after the

change ("H" to "L" or "L" to "H") in the next counter output to be executed. The counting operation at the gate input is done the same as in mode 2.

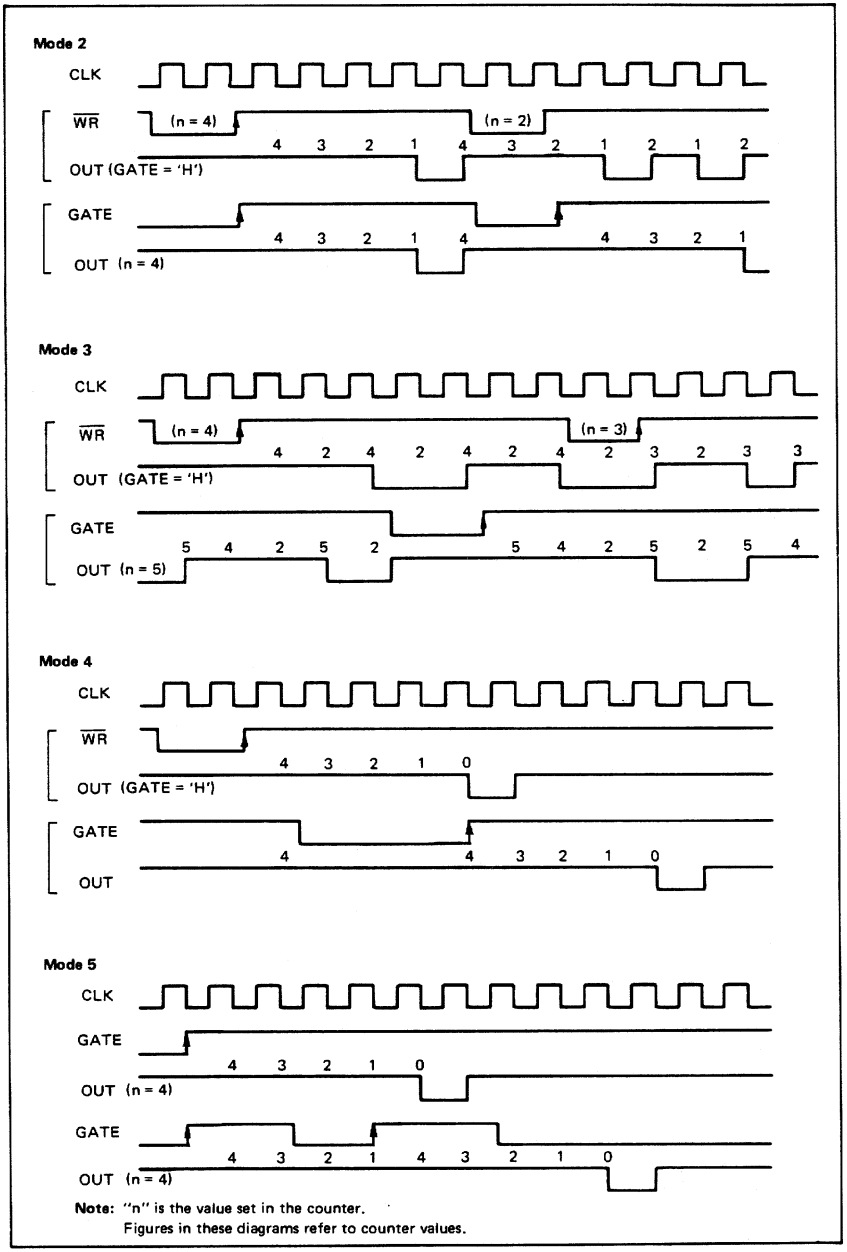
- **Mode 4 (software trigger strobe)**
  The counter output is switched to "H" level by the mode setting. Counting is started in the same way as described for mode 0. A single "L" pulse equivalent to one clock width is generated at the counter output when the terminal count is reached.
  This mode differs from 2 in that the "L" level output appears one clock earlier in mode 2, and that pulses are not repeated in mode 4. Counting is stopped when the gate input is switched to "L" level, and restarted from the set count value when switched back to "H" level.

- **Mode 5 (hardware trigger strobe)**
  The counter output is switched to "H" level by the mode setting. Counting is started, and the gate input used, in the same way as in mode 1.
  The counter output is identical to the mode 4 output.
  The various roles of the gate input signals in the above modes are summarized in the following table.

| Mode \ Gate | "L" Level Falling Edge | Rising Edge | "H" Level |
|---|---|---|---|
| 0 | Counting not possible | | Counting possible |
| 1 | | (1) Start of counting<br>(2) Retriggering | |
| 2 | (1) Counting not possible<br>(2) Counter output forced to "H" level | Start of counting | Counting possible |
| 3 | (1) Counting not possible<br>(2) Counter output forced to "H" level | Start of counting | Counting possible |
| 4 | Counting not possible | | Counting possible |
| 5 | | (1) Start of counting<br>(2) Retriggering | |



■ I/O-MSM82C53-2RS/GS/JS ■




**Note:** "n" is the value set in the counter.
Figures in these diagrams refer to counter values.

**Reading of Counter Values**

All 82C53 counting is down-counting, the counting being in steps of 2 in mode 3. Counter values can be read during counting by (1) direct reading, and (2) counter latching ("read on the fly").

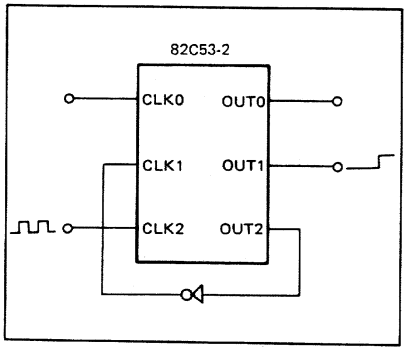
- **Direct reading**
  Counter values can be read by direct reading operations.
  Since the counter value read according to the timing of the $\overline{RD}$ and CLK signals is not guaranteed, it is necessary to stop the counting by a gate input signal, or to interrupt the clock input temporarily by an external circuit to ensure that the counter value is correctly read.

- **Counter latching**
  In this method, the counter value is latched by writing a counter latch command, thereby enabling a stable value to be read without effecting the counting in any way at all. An example of a counter latching program is given below.
  Counter latching executed for counter #1 (Read/Load 2-byte setting)

```
MVI A  0 1 0 0 x x x x
                └── Denotes counter latching
OUT n3 ──┬── Write in control word address (n3)
         └── The counter value at this point is latched
IN n1  ──┬── Reading of the LSB of the counter value latched from counter #1.
         └── n1: Counter #1 address
MOV B, A
IN n1  ──── Reading of MSB from counter #1.
MOV C, A
```

**Example of Practical Application**

- **82C53 used as a 32-bit counter.**




Use counter #1 and counter #2
Counter #1: mode 0, upper order 16-bit counter value
Counter #2: mode 2, lower order 16-bit counter value
This setting enables counting up to a maximum of $2^{32}$

# E

# Customer Communication

For your convenience, this appendix contains forms to help you gather the information necessary to help us solve your technical problems and a form you can use to comment on the product documentation. When you contact us, we need the information on the Technical Support Form and the configuration form, if your manual contains one, about your system configuration to answer your questions as quickly as possible.

National Instruments has technical assistance through electronic, fax, and telephone systems to quickly provide the information you need. Our electronic services include a bulletin board service, an FTP site, a fax-on-demand system, and e-mail support. If you have a hardware or software problem, first try the electronic support systems. If the information available on these systems does not answer your questions, we offer fax and telephone support through our technical support centers, which are staffed by applications engineers.

## Electronic Services

### Bulletin Board Support

National Instruments has BBS and FTP sites dedicated for 24-hour support with a collection of files and documents to answer most common customer questions. From these sites, you can also download the latest instrument drivers, updates, and example programs. For recorded instructions on how to use the bulletin board and FTP services and for BBS automated information, call 512 795 6990. You can access these services at:

United States: 512 794 5422
Up to 14,400 baud, 8 data bits, 1 stop bit, no parity

United Kingdom: 01635 551422
Up to 9,600 baud, 8 data bits, 1 stop bit, no parity

France: 01 48 65 15 59
Up to 9,600 baud, 8 data bits, 1 stop bit, no parity

### FTP Support

To access our FTP site, log on to our Internet host, `ftp.natinst.com`, as `anonymous` and use your Internet address, such as `joesmith@anywhere.com`, as your password. The support files and documents are located in the `/support` directories.

## Fax-on-Demand Support

Fax-on-Demand is a 24-hour information retrieval system containing a library of documents on a wide range of technical information. You can access Fax-on-Demand from a touch-tone telephone at 512 418 1111.

## E-Mail Support (Currently USA Only)

You can submit technical support questions to the applications engineering team through e-mail at the Internet address listed below. Remember to include your name, address, and phone number so we can contact you with solutions and suggestions.

`support@natinst.com`

# Telephone and Fax Support

National Instruments has branch offices all over the world. Use the list below to find the technical support number for your country. If there is no National Instruments office in your country, contact the source from which you purchased your software to obtain support.

| Country | Telephone | Fax |
|---|---|---|
| Australia | 03 9879 5166 | 03 9879 6277 |
| Austria | 0662 45 79 90 0 | 0662 45 79 90 19 |
| Belgium | 02 757 00 20 | 02 757 03 11 |
| Brazil | 011 288 3336 | 011 288 8528 |
| Canada (Ontario) | 905 785 0085 | 905 785 0086 |
| Canada (Quebec) | 514 694 8521 | 514 694 4399 |
| Denmark | 45 76 26 00 | 45 76 26 02 |
| Finland | 09 725 725 11 | 09 725 725 55 |
| France | 01 48 14 24 24 | 01 48 14 24 14 |
| Germany | 089 741 31 30 | 089 714 60 35 |
| Hong Kong | 2645 3186 | 2686 8505 |
| Israel | 03 6120092 | 03 6120095 |
| Italy | 02 413091 | 02 41309215 |
| Japan | 03 5472 2970 | 03 5472 2977 |
| Korea | 02 596 7456 | 02 596 7455 |
| Mexico | 5 520 2635 | 5 520 3282 |
| Netherlands | 0348 433466 | 0348 430673 |
| Norway | 32 84 84 00 | 32 84 86 00 |
| Singapore | 2265886 | 2265887 |
| Spain | 91 640 0085 | 91 640 0533 |
| Sweden | 08 730 49 70 | 08 730 43 70 |
| Switzerland | 056 200 51 51 | 056 200 51 55 |
| Taiwan | 02 377 1200 | 02 737 4644 |
| United Kingdom | 01635 523545 | 01635 523154 |
| United States | 512 795 8248 | 512 794 5678 |

# Technical Support Form

Photocopy this form and update it each time you make changes to your software or hardware, and use the completed copy of this form as a reference for your current configuration. Completing this form accurately before contacting National Instruments for technical support helps our applications engineers answer your questions more efficiently.

If you are using any National Instruments hardware or software products related to this problem, include the configuration forms from their user manuals. Include additional pages if necessary.

Name _____

Company _____

Address _____

_____

Fax ( ___ ) _____Phone ( ___ ) _____

Computer brand_____ Model _____ Processor_____

Operating system (include version number) _____

Clock speed _____MHz RAM _____MB Display adapter _____

Mouse ___yes ___no Other adapters installed_____

Hard disk capacity _____MB Brand_____

Instruments used _____

_____

National Instruments hardware product model _____ Revision _____

Configuration _____

National Instruments software product _____ Version _____

Configuration _____

The problem is: _____

_____

_____

_____

_____

List any error messages: _____

_____

_____

The following steps reproduce the problem: _____

_____

_____

_____

_____

# PCI-DIO-96/PXI-6508/PCI-6503 Hardware and Software Configuration Form

Record the settings and revisions of your hardware and software on the line to the right of each item. Complete a new copy of this form each time you revise your software or hardware configuration, and use this form as a reference for your current configuration. Completing this form accurately before contacting National Instruments for technical support helps our applications engineers answer your questions more efficiently.

## National Instruments Products

PCI-DIO-96/PXI-6508/PCI-6503 board _____

Board serial number _____

Board base memory address _____

Programming choice (NI-DAQ, LabVIEW, LabWindows/CVI, or other) _____

Software version _____

Other National Instruments boards in system _____

Base memory address of other boards _____

Interrupt level of other boards _____

## Other Products

Computer make and model _____

Microprocessor _____

Clock frequency or speed _____

Type of video board installed _____

Operating system and version _____

Operating system mode _____

Programming language _____

Programming language version _____

Other boards in system _____

Base memory address of other boards _____

Interrupt level of other boards _____

# Documentation Comment Form

National Instruments encourages you to comment on the documentation supplied with our products. This information helps us provide quality products to meet your needs.

**Title:** *PCI-DIO-96/PXI-6508/PCI-6503 User Manual*

**Edition Date:** March 1998

**Part Number:** 320938C-01

Please comment on the completeness, clarity, and organization of the manual.

_____

_____

_____

_____

_____

_____

_____

If you find errors in the manual, please record the page numbers and describe the errors.

_____

_____

_____

_____

_____

_____

_____

Thank you for your help.

Name _____

Title _____

Company _____

Address _____

_____

E-Mail Address _____

Phone ( ___ ) _____ Fax ( ___ ) _____

**Mail to:** Technical Publications
National Instruments Corporation
6504 Bridge Point Parkway
Austin, Texas 78730-5039

**Fax to:** Technical Publications
National Instruments Corporation
512 794 5678

# Glossary

| Prefix | Meanings | Value |
|---|---|---|
| p- | pico | $10^{-12}$ |
| n- | nano- | $10^{-9}$ |
| µ- | micro- | $10^{-6}$ |
| m- | milli- | $10^{-3}$ |
| k- | kilo- | $10^{3}$ |
| M- | mega- | $10^{6}$ |
| G- | giga- | $10^{9}$ |

## Numbers/Symbols

| | |
|---|---|
| ° | degrees |
| > | greater than |
| ≥ | greater than or equal to |
| < | less than |
| – | negative of, or minus |
| Ω | ohms |
| / | per |
| % | percent |
| ± | plus or minus |
| + | positive of, or plus |
| +5 V | +5 Volts signal |

# A

| | |
|---|---|
| A | amperes |
| ACK* | acknowledge input signal |
| AIRQ0 | PPI A port A interrupt enable bit |
| AIRQ1 | PPI A port B interrupt enable bit |
| ANSI | American National Standards Institute |
| APA | PPI A port A |
| APB | PPI A port B |
| APC | PPI A port C |
| ASIC | Application Specific Integrated Circuit |
| AWG | American Wire Gauge |

# B

| | |
|---|---|
| BCD | binary coded decimal |
| BIRQ0 | PPI B port A interrupt enable bit |
| BIRQ1 | PPI B port B interrupt enable bit |
| BPA | PPI B port A |
| BPB | PPI B port B |
| BPC | PPI B port C |

# C

| | |
|---|---|
| C | Celsius |
| CIRQ0 | PPI C port A interrupt enable bit |
| CIRQ1 | PPI C port B interrupt enable bit |

| | |
|---|---|
| cm | centimeters |
| CompactPCI | refers to the core specification defined by the PCI Industrial Computer Manufacturer's Group (PICMG) |
| CPA | PPI C port A |
| CPB | PPI C port B |
| CPC | PPI C port C |
| CTR1 | counter select bit |
| CTRIRQ | counter interrupt enable bit |

## D

| | |
|---|---|
| DAQ | a system that uses the personal computer to collect, measure, and generate electrical signals |
| DI | digital input |
| DIO | digital input/output |
| DIRQ0 | PPI D port A interrupt enable bit |
| DIRQ1 | PPI D port B interrupt enable bit |
| DMA | direct memory access—a method by which data can be transferred to/from computer memory from/to a device or memory on the bus while the processor does something else. DMA is the fastest method of transferring data to/from computer memory. |
| DO | digital output |
| DPA | PPI D port A |
| DPB | PPI D port B |
| DPC | PPI D port C |

## F

| | |
|---|---|
| ft | feet |

## G

| | |
|---|---|
| GND | ground signal |

## H

| | |
|---|---|
| hex | hexadecimal |

## I

| | |
|---|---|
| IBF | input buffer full signal |
| in. | inches |
| INTE1 | port A output interrupt enable bit |
| INTE2 | port A input interrupt enable bit |
| INTEA | port A interrupt enable bit |
| INTEB | port B interrupt enable bit |
| INTEN | interrupt enable bit |
| INTRA | port A interrupt request status |
| INTRB | port B interrupt request status |
| I/O | input/output |

## L

| | |
|---|---|
| LED | light-emitting diode |
| LSB | least significant bit |

## M

| | |
|---|---|
| m | meters |

| | |
|---|---|
| max | maximum |
| MB | megabytes of memory |
| min. | minutes |
| min | minimum |
| MSB | most significant bit |

## O

| | |
|---|---|
| OBF* | output buffer full signal |

## P

| | |
|---|---|
| PA, PB, PC <0..7> | port A, B, or C 0 through 7 lines |
| PCI | Peripheral Component Interconnect—a high-performance expansion bus architecture originally developed by Intel to replace ISA and EISA. It is achieving widespread acceptance as a standard for PCs and workstations; it offers a theoretical maximum transfer rate of 132 Mbytes/s. |
| port | a digital port, consisting of four or eight lines of digital input and/or output |
| PPI | programmable peripheral interface |
| PXI | PCI eXtensions for Instrumentation. PXI is an open specification that builds off the CompactPCI specification by adding instrumentation-specific features. |

## R

| | |
|---|---|
| RD* | read signal |

## S

| | |
|---|---|
| S | samples |
| s | seconds |

| | |
|---|---|
| SCXI | Signal Conditioning eXtensions for Instrumentation—the National Instruments product line for conditioning low-level signals within an external chassis near sensors so only high-level signals are sent to DAQ boards in the noisy PC environment |
| signal conditioning | the manipulation of signals to prepare them for digitizing |
| STB | strobe input signal |

## T

| | |
|---|---|
| TTL | transistor-transistor logic |
| typ | typical |

## V

| | |
|---|---|
| V | volts |
| $V_{cc}$ | supply voltage; for example, the voltage a computer supplies to its plug-in devices |
| VDC | volts direct current |
| VI | virtual instrument—a combination of hardware and/or software elements, typically used with a PC, that has the functionality of a classic standalone instrument |
| $V_{in}$ | input voltage |

## W

| | |
|---|---|
| W | watts |
| WRT* | write signal |

# Index

## Numbers


+5 V supply pin
  connecting directly to ground or other voltage source (warning), 3-12
  PCI-6503 (table), 3-8
  PCI-DIO-96 and PXI-6508 (table), 3-5
  self-resetting fuse, 3-12
82C53 Programmable Interval Timer
  data sheet, D-1
  theory of operation, 4-3
82C53 programming considerations, B-32 to B-33
  general information, B-33
  interrupt programming example, B-33
82C53 Register Group
  control word format (figure), B-5
  Interrupt Clear Register, B-10
  Interrupt Control Register 1, B-7 to B-8
  Interrupt Control Register 2, B-9
  register map (table), B-2
82C55A Programmable peripheral Interface
  data sheet, C-1
  theory of operation, 4-3
82C55A programming considerations
  examples
    interrupt programming, B-31 to B-32
    Mode 0 basic I/O, B-18
    Mode 1 strobed input, B-22
    Mode 1 strobed output, B-26
    Mode 2 bidirectional bus, B-30
  modes of operation, B-15 to B-19
    Mode 0, B-15
    Mode 1, B-15
    Mode 2, B-16
    single bit set/reset feature, B-16
82C55A Register Group
  control word formats (figure), B-4
  description, B-3 to B-5
  Port C set/reset control words (table), B-5
  register map (table), B-2


## A


ACK* signal
  description (table), 3-16
  Mode 1 output timing (figure), 3-18
  Mode 2 bidirectional timing (figure), 3-19
AIRQ0 bit, B-8
AIRQ1 bit, B-8
APA<7..0> signal (table), 3-5
APB<7..0> signal (table), 3-5
APC<7..0> signal (table), 3-5


## B


BIRQ0 bit, B-8
BIRQ1 bit, B-8
bit descriptions. *See also* Port C status-word bit definitions.
  AIRQ0, B-8
  AIRQ1, B-8
  BIRQ0, B-8
  BIRQ1, B-8
  CIRQ0, B-7, B-31
  CIRQ1, B-7
  CTR1, B-9
  CTRIRQ, B-9
  DIRQ0, B-7
  DIRQ1, B-7
  INTEN, B-9, B-31
board configuration, 2-2
BPA<7..0> signal (table), 3-5

BPB<7..0> signal (table), 3-5
BPC<7..0> signal (table), 3-5
bulletin board support, E-1
bus interface specifications, A-3

## C

cabling, custom, 1-6
CIRQ0 bit
  description, B-7
  interrupt handling, B-31
CIRQ1 bit, B-7
CompactPCI, using with PXI-compatible products, 1-2
ComponentWorks software, 1-3
configuration
  board configuration, 2-2
  configuration program for Macintosh PCI initialization, B-14 to B-15
control words
  82C53 Register Group control word format (figure), B-5
  82C55A Register Group
    control word formats (figure), B-4
    Port C set/reset control words (table), B-5
  Mode 1 strobed input
    Port A configuration (figure), B-18
    Port B configuration (figure), B-19
  Mode 1 strobed output
    Port A configuration (figure), B-23
    Port B configuration (figure), B-23
  Mode 2 bidirectional bus (figure), B-27
CPA<7..0> signal (table), 3-6
CPB<7..0> signal (table), 3-6
CPC<7..0> signal (table), 3-5
CTR1 bit, B-9
CTRIRQ bit, B-9
custom cabling, 1-6
customer communication, *xii*, E-1 to E-2

## D

DATA signal
  description (table), 3-16
  Mode 1 input timing (figure), 3-17
  Mode 1 output timing (figure), 3-18
  Mode 2 bidirectional timing (figure), 3-19
digital I/O power-up state selection, 3-12 to 3-15
  high DIO power-up state, 3-12 to 3-13
  low DIO power-up state, 3-14 to 3-15
digital I/O signal connections, 3-10 to 3-11
  block diagram, 3-11
  specifications and ratings, 3-10 to 3-11
digital I/O specifications, A-1
digital logic levels, A-1 to A-2
  input signals (table), A-1 to A-2
  output signals, A-2
DIRQ0 bit, B-7
DIRQ1 bit, B-7
documentation
  conventions used in manual, *x* to *xi*
  National Instruments documentation, *xi* to *xii*
  organization of manual, *ix* to *x*
  related documentation, *xii*
DPA<7..0> signal (table), 3-6
DPB<7..0> signal (table), 3-6
DPC<7..0> signal (table), 3-6

## E

electronic support services, E-1 to E-2
e-mail support, E-2
environment specifications, A-4
equipment, optional, 1-5

## F

fax and telephone support numbers, E-2
Fax-on-Demand support, E-2
FTP support, E-1
fuse, self-resetting, 3-12

## G

Get_DAQ_device_Info call, for simple accesses, B-13
GND signal
PCI-6503 (table), 3-8
PCI-DIO-96 and PXI-6508 (table), 3-5

## H

hardware installation
PCI-DIO-96 or PCI-6503, 2-1 to 2-2
PXI-6508, 2-2

## I

IBF signal
description (table), 3-16
Mode 1 input timing (figure), 3-17
Mode 2 bidirectional timing (figure), 3-19
IBFA bit
Mode 1 strobed input, B-20
Mode 2 bidirectional bus, B-28
IBFB bit, B-20
installation
hardware, 2-1 to 2-2
software, 2-1
unpacking the PCI-DIO-96, 1-6
INTE1 bit
interrupt handling, B-31
Mode 2 bidirectional bus, B-28
INTE2 bit
interrupt handling, B-31
Mode 2 bidirectional bus, B-28
INTEA bit
interrupt handling, B-31
Mode 1 strobed input, B-20
Mode 1 strobed output, B-24
INTEB bit
interrupt handling, B-31
Mode 1 strobed input, B-20
Mode 1 strobed output, B-24
INTEN bit
description, B-9
interrupt handling, B-31
interrupt control circuitry
block diagram, 4-4
theory of operation, 4-3
Interrupt Control Register Group
Interrupt Clear Register, B-10
Interrupt Control Register 1, B-7 to B-8
Interrupt Control Register 2, B-9
register map (table), B-2
interrupt generation for Macintosh computers
developing your own method, B-14 to B-15
performing simple accesses, B-13
using NI-DAQ software, B-13
interrupt handling, B-31 to B-32
82C53 programming example, B-33
82C55A programming examples, B-31 to B-32
requirements for, B-31
INTR signal
description (table), 3-16
Mode 1 input timing (figure), 3-17
Mode 1 output timing (figure), 3-18
Mode 2 bidirectional timing (figure), 3-19
INTRA bit
Mode 1 strobed input, B-20
Mode 1 strobed output, B-24
Mode 2 bidirectional bus, B-28
INTRB bit
Mode 1 strobed input, B-21
Mode 1 strobed output, B-25

I/O bit
Mode 1 strobed input, B-20
Mode 1 strobed output, B-24
Mode 2 bidirectional bus, B-29
I/O connector
cable assembly connector pinouts (figure)
pins 1 through 50, 3-3
pins 51 through 100, 3-4
exceeding maximum ratings (warning), 3-1
pin assignments
PCI-6503 (figure), 3-7
PCI-DIO-96 and PXI-6508 (figure), 3-2
Port C pin assignments
Mode 1 input (figure), B-22
Mode 1 output (figure), B-26
signal connection descriptions
PCI-6503 (table), 3-8
PCI-DIO-96 and PXI-6508 (table), 3-5 to 3-6

## L

LabVIEW software, 1-3
LabWindows/CVI software, 1-3

## M

manual. *See* documentation.
MITE ASIC. *See* PCI MITE ASIC.
Mode 0 basic I/O
I/O configurations (table), B-17
programming considerations, B-15
programming example, B-18
Mode 1 input
interrupt programming example, B-32
programming considerations, B-15
strobed input programming considerations, B-18 to B-22
control word to configure Port A (figure), B-18
control word to configure Port B (figure), B-19
Port C pin assignments on I/O connector (figure), B-22
Port C status-word bit definitions for input, B-20 to B-22
programming example, B-22
timing (figure), 3-17
Mode 1 output
interrupt programming example, B-32
programming considerations, B-15
strobed output programming considerations, B-22 to B-26
control word to configure Port A (figure), B-23
control word to configure Port B (figure), B-23
Port C pin assignments on I/O connector (figure), B-26
Port C status-word bit definitions for output, B-24 to B-26
programming example, B-26
timing (figure), 3-18
Mode 2 bidirectional bus
interrupt programming example, B-32
programming considerations, B-26 to B-30
control word to configure Port A (figure), B-27
Port C pin assignments on I/O connector, B-30
Port C status-word bit definitions, B-28 to B-30
programming example, B-30
timing (figure), 3-19

## N

NI-DAQ driver software, 1-3 to 1-4
  interrupt generation
    Get_DAQ_device_Info call for simple accesses, B-13
    PCI initialization for Macintosh, B-13
  overview, 1-3 to 1-4
  relationship to programming environment (figure), 1-4

## O

OBF* signal
  description (table), 3-16
  Mode 1 output timing (figure), 3-18
  Mode 2 bidirectional timing (figure), 3-19
OBFA* bit
  Mode 1 strobed output, B-24
  Mode 2 bidirectional bus, B-28
OBFB* bit, B-24
operation of DIO board. *See* theory of operation.
optional equipment, 1-5

## P

PA<7..0> signal (table), 3-8
PB<7..0> signal (table), 3-8
PC<7..0> signal (table), 3-8
PCI initialization
  for Macintosh, B-13 to B-15
    developing your own configuration program, B-14 to B-15
    performing simple accesses, B-13
    using NI-DAQ and driver toolkit, B-13
  for PCs, B-11 to B-12
PCI interface circuitry, 4-2
PCI local bus
  PCI Local Bus Specification, Revision 2.1, B-10
  programming considerations, B-10 to B-11
PCI MITE ASIC
  configuration
    programming, B-11 to B-12
    pseudocode for, B-12
  overview, 4-2
PCI-DIO-96/PXI-6508/PCI-6503 board
  block diagram, 4-2
  configuration, 2-2
  custom cabling, 1-6
  optional equipment, 1-5
  overview, 1-1
  requirements for getting started, 1-2
  software programming choices, 1-3 to 1-5
    National Instruments application software, 1-3
    NI-DAQ driver software, 1-3 to 1-4
    register-level programming, 1-5
  unpacking, 1-6
physical specifications, A-4
Port C pin assignments
  correlation between mode and handshaking terminology (table), 3-9
  I/O connector
    Mode 1 input (figure), B-22
    Mode 1 output (figure), B-26
    Mode 2 bidirectional bus, B-30
  mode configuration, 3-9
  overview, 3-9
Port C set/reset control words (table), B-5

Port C status-word bit definitions
Mode 1 strobed input, B-20 to B-22
IBFA, B-20
IBFB, B-20
INTEA, B-20
INTEB, B-20
INTRA, B-20
INTRB, B-21
I/O, B-20
Mode 1 strobed output, B-24 to B-26
INTEA, B-24
INTEB, B-24
INTRA, B-24
INTRB, B-25
I/O, B-24
OBFA*, B-24
OBFB*, B-24
Mode 2 bidirectional bus, B-28 to B-30
IBFA, B-28
INTE1, B-28
INTE2, B-28
INTRA, B-28
I/O, B-29
OBFA*, B-28
power connections, 3-12
power requirement specifications, A-3
power-up state selection. *See* digital I/O power-up state selection.
programming. *See* register-level programming.
PXI-compatible products, using with CompactPCI, 1-2

## R

RD* signal
description (table), 3-16
Mode 1 input timing (figure), 3-17
Mode 2 bidirectional timing (figure), 3-19
register-level programming, B-10 to B-33
82C53 programming considerations, B-32 to B-33
82C55A modes of operation, B-15 to B-19
Mode 0, B-15
Mode 1, B-15
Mode 2, B-16
single bit set/reset feature, B-16
examples, B-11 to B-15
common terms used in examples, B-11
developing your own configuration program, B-14 to B-15
Mode 0 basic I/O, B-18
Mode 1 strobed input, B-22
Mode 1 strobed output, B-26
Mode 2 bidirectional bus, B-30
PCI initialization for Macintosh, B-13 to B-15
PCI initialization for PCs, B-11 to B-12
interrupt handling, B-31-32
82C53 programming example, B-33
82C55A programming examples, B-32
developing your own interrupt method, B-14 to B-15
Mode 1 strobed input, B-18 to B-22
control word to configure Port A (figure), B-18
control word to configure Port B (figure), B-19
Port C pin assignments on I/O connector (figure), B-22
Port C status-word bit definitions for input, B-20 to B-22
programming example, B-22

Mode 1 strobed output, B-22 to B-26
control word to configure Port A (figure), B-23
control word to configure Port B (figure), B-23
Port C pin assignments on I/O connector (figure), B-26
Port C status-word bit definitions for output, B-24 to B-26
programming example, B-26
Mode 2 bidirectional bus, B-26 to B-30
control word to configure Port A (figure), B-27
Port C pin assignments on I/O connector, B-30
Port C status-word bit definitions, B-28 to B-30
programming example, B-30
PCI initialization for Macintosh
developing your own configuration program, B-14 to B-15
performing simple accesses, B-13
programming options, B-13
using NI-DAQ and driver toolkit, B-13
PCI initialization for PCs, B-11 to B-12
PCI local bus, B-10 to B-11
using other software instead, 1-5
registers
82C53 Register Group
control word format (figure), B-5
Interrupt Clear Register, B-10
Interrupt Control Register 1, B-7 to B-8
Interrupt Control Register 2, B-9
register map (table), B-2
82C55A Register Group
control word formats (figure), B-4
description, B-3 to B-5
Port C set/reset control words (table), B-5
register map (table), B-2
overview, A-1
register map (table), B-2
terminology
clarification for programming examples, B-11
correlation between mode and handshaking terminology (table), 3-9


## S


self-resetting fuse, 3-12
signal connections
digital I/O signal connections, 3-10 to 3-11
block diagram, 3-11
digital logic levels, A-1 to A-2
input signals (table), A-1 to A-2
output signals, A-2
I/O connector
cable assembly connector pinouts (figure)
pins 1 through 50, 3-3
pins 51 through 100, 3-4
exceeding maximum ratings (warning), 3-1
PCI-6503 connector pin assignments (figure), 3-7
PCI-DIO-96 and PXI-6508 connector pin assignments (figure), 3-2
Port C pin assignments, 3-9
power connections, 3-12

signal descriptions (table)
PCI-6503, 3-8
PCI-DIO-96 and PXI-6508, 3-5 to 3-6
timing specifications, 3-15 to 3-19
Mode 1 input timing, 3-17
Mode 1 output timing, 3-18
Mode 2 bidirectional timing, 3-19
signal names used in timing diagrams (table), 3-16
software installation, 2-1
software programming choices, 1-3 to 1-5
National Instruments application software, 1-3
NI-DAQ driver software, 1-3 to 1-4
register-level programming, 1-5
specifications
bus interface, A-3
digital I/O, A-1
digital logic levels, A-1 to A-2
input signals (table), A-1 to A-2
output signals, A-2
environment, A-4
physical, A-4
power requirement, A-3
transfer rates, A-3
STB* signal
description (table), 3-16
Mode 1 input timing (figure), 3-17
Mode 2 bidirectional timing (figure), 3-19


## T


technical support, E-1 to B-2
telephone and fax support numbers, E-2
terminology related to registers
correlation between mode and handshaking terminology (table), 3-9
terms used in programming examples, B-11
theory of operation
82C53 Programmable Interval Timer, 4-3
82C55A Programmable peripheral Interface, 4-3
block diagram of PCI-DIO-96/PXI-6508/PCI-6503, 4-2
interrupt control circuitry, 4-3
block diagram, 4-4
PCI interface circuitry, 4-2
timing specifications, 3-15 to 3-19
Mode 1 input timing, 3-17
Mode 1 output timing, 3-18
Mode 2 bidirectional timing, 3-19
signal names used in timing diagrams (table), 3-16
transfer rate specifications, A-3


## U


unpacking the DIO board, 1-6


## W


WR* signal
description (table), 3-16
Mode 1 output timing (figure), 3-18
Mode 2 bidirectional timing (figure), 3-19